DEFENCE RESEARCH ESTABLISHMENT
CENTRE DE RECHERCHES POUR LA DÉFENSE
VALCARTIER, QUÉBEC

DREV TR 1999-153
Unlimited Distribution / Distribution illimitée

DEFINITION OF A TEST BED FOR THE EVALUATION OF REAL-TIME MULTI-SOURCE DATA FUSION SYSTEMS

by
J.M.J. Roy

September / septembre 2000

Approved by / approuvé par

D. J. Marrissey

Section Head / Chef de section

9 Aug 00

Date

102-00859/

ABSTRACT

Multi-Source Data Fusion (MSDF) is about tactical picture compilation. An MSDF system processes the data and information reported by multiple dissimilar sources in order to correctly and quickly derive the best estimates of the current and future kinematics properties for each hypothesized (or perceived) entity in the operational environment, and to develop inferences as to the identity and key attributes of these entities. From a research and development perspective, MSDF system developers need capabilities that allow them to assess if a proposed MSDF System is working properly, as expected. Tools are required to support the developer in his quantitative assessment of the performance of the MSDF algorithms and techniques. However, the MSDF systems foreseen to support command teams onboard Canadian warships must be real-time systems. The correctness of the final implementation of a real-time system is not only a function of the accuracy of the computations performed by the system, but also a function of the time at which the results of the computations are produced. This document gives a detailed description of a test bed that will help to establish the real-time requirements of MSDF systems on speed, responsiveness, predictability, timeliness, etc. At the same time, the proposed MSDF test bed will also provide a capability to evaluate various measures of performance for these MSDF Systems.

RÉSUMÉ

L'objet de la fusion de données multi-sources (FDMS) est la génération de l'image tactique. Un système de FDMS traite les données et l'information de plusieurs sources dans le but de dériver la meilleure estimation des propriétés de cinématique de chaque entité perçue dans l'environnement et d'inférer l'identité et les attributs-clés de ces entités. D'un point de vue recherche et développement, les concepteurs de systèmes de FDMS ont besoin de moyens leur permettant d'évaluer si le système de FDMS proposé fonctionne convenablement comme prévu. Des outils sont nécessaires pour appuyer le concepteur dans son évaluation quantitative de la performance des algorithmes et techniques de FDMS. Cependant, les systèmes de FDMS envisagés pour soutenir les équipes de commandement à bord des navires de guerre canadiens doivent être des systèmes en temps réel. L'exactitude de l'implantation finale d'un système en temps réel est non seulement fonction de la précision des calculs effectués par le système, mais également du moment où les résultats de ces calculs sont produits. Ce document présente la description détaillée d'un banc d'essai qui aidera à établir les besoins en temps réel des systèmes de FDMS concernant la vitesse, la bonne réaction, la prévisibilité, l'à-propos, etc. En même temps, le banc d'essai proposé fournira aussi un moyen d'évaluer différentes mesures de la performance pour ces systèmes de FDMS.

WARNING NOTICE

The information contained herein is proprietary to Her Majesty and is provided to the recipient on the understanding that it will be used for information and evaluation purposes only. Any commercial use, including use for manufacture, is prohibited. Release to third parties of this publication or of information contained herein is prohibited without the prior written consent of DND Canada.

© Her Majesty the Queen in Right of Canada as represented by the Minister of National Defence, 2000

TABLE OF CONTENTS

ABSTRACT/RÉSUMÉ i
EXECUTIVE SUMMARY v
LIST OF ACRONYMS vii
1.0 INTRODUCTION 1
2.0 MSDF TEST BED SYSTEM OVERVIEW 6
2.1 Definitions 7
2.2 Development Environment 9
2.3 Stimulation Module 9
2.4 MSDF System Module 10
2.5 Performance Analysis Database (PADB) 12
2.6 Performance Evaluation Module 12
2.7 Human Computer Interface 12
3.0 MSDF TEST BED STATES AND MODES 14
3.1 States/Modes Transitions 14
3.2 Control State 15
3.3 Control State / Development Environment Mode 15
3.4 Control State / Test Bed Manager Mode 16
3.5 On-Line State 16
3.6 On-Line State / Real-Time Mode 17
3.7 On-Line State / Step-by-Step Mode 18
3.8 Off-Line State 20
3.9 Off-Line State / Replay Mode 21
3.10 Off-Line State / Performance Evaluation Mode 22
4.0 STIMULATION MODULE 24
4.1 Stimulation Module Inputs / Outputs 24
4.2 Pre-Saved Stimulation Files 24
4.3 Data Driver 25

TABLE OF CONTENTS (cont'd)

5.0 PERFORMANCE ANALYSIS DATABASE (PADB) 29
5.1 PADB in the On-Line State 29
5.2 PADB in the Off-Line State / Replay Mode 32
5.3 PADB in the Off-Line State / Performance Evaluation Mode 33
6.0 HUMAN COMPUTER INTERFACE (HCI) MODULE 36
6.1 HCI Module Inputs/Outputs 36
6.2 Manager Display 37
6.3 Analytical Display 48
6.4 Performance Evaluation Display 62
7.0 PERFORMANCE EVALUATION MODULE 64
7.1 PE Module Inputs/Outputs 66
7.2 MSDF Measures Of Performance (MOPs) 68
7.3 Real-Time System Performance Measures 70
7.4 MSDF System Tuning 70
8.0 CONCLUSION 72
9.0 ACKNOWLEDGEMENTS 75
10.0 REFERENCES 76
FIGURES 1 to 8
APPENDIX A - Examples of Black and White Box Data 77

EXECUTIVE SUMMARY

In a maritime context, the graphic and textual depiction of any information that may affect the Maritime Commander's decision making in an area of interest is called the Recognized Maritime Picture (RMP). More precisely, the term Maritime Tactical Picture (MTP) is typically used for a Tactical Coordination Center (TCC) afloat. Tactical commanders at all levels will require a temporary and often moving MTP of their battle space. The information necessary to compile the MTP will come from many sources. In carrying out their missions, the crews of the HALIFAX/IROQUOIS class ships are typically bombarded with sensor and link information. All these data required to compile the MTP must eventually be easily accessible and displayed on one system, so that the commander and his staff can view the data from one display. In general terms, the data will need to be combined into a single picture and displayed in the best manner to facilitate the command and control process. These are clearly data fusion issues.

Data fusion is the collation of related information from many sources into information portrayed as coming from a single source; it is the key MTP enabler. Unfortunately, the current combat systems within the Canadian Navy do not contain much data fusion capability as part of the processing functions. At present, the fusion of information coming from multiple sources is being manually performed by the operators. This process of manually and mentally re-plotting information is very laborious, complex, time consuming and prone to error. Mental and manual data fusion must be replaced by automated data fusion wherever it makes sense and is possible. As a subset of data fusion, Multi-Source Data Fusion (MSDF) is about tactical picture compilation. An MSDF system processes the data reported by multiple sources in order to derive the best estimates of the kinematic properties for each perceived entity in the environment, and to infer the identity and key attributes of these entities. Defence Research Establishment Valcartier (DREV) has been working for over a decade to develop technologies to enable Canada's warships to dynamically and automatically obtain an image of a tactical situation. The scope of this research is within the time frame of the HALIFAX class mid-life refit and the spinoffs will also be of direct benefit to the IROQUOIS class and its potential replacement.

Tools are required to support the developer in quantitatively assessing the performance of the MSDF algorithms and techniques. However, the MSDF systems foreseen to support command teams onboard ships must be real-time systems. The correctness of the final implementation of a real-time system is not only a function of the accuracy of the computations performed by the system, but also a function of the time at which the results of the computations are produced. This document describes a test bed that will help to establish the real-time requirements of MSDF systems on speed, responsiveness, predictability, timeliness, etc. At the same time, the proposed MSDF test bed will also provide a capability to evaluate various measures of performance for these MSDF Systems.

LIST OF ACRONYMS

AOU	Area Of Uncertainty
ARO	Auxiliary ReadOut
ASCACT	Advanced Shipboard Command and Control Technology
AWW	Above Water Warfare
CASE_ATTI	Concept Analysis and Simulation Environment for Automatic Target Tracking and Identification
C2	Command and Control
CCRO	Close Control ReadOut
CPF	Canadian Patrol Frigate
DARO	Data Amplification ReadOut
DREV	Defence Research Establishment Valcartier
ESM	Electronic Surveillance Measures
FDMS	Fusion de données multi-sources
FIFO	First In First Out
GID	Generic Identification
HCI	Human Computer Interface
ID	Identification
IFF	Identification Friend or Foe
MHT	Multiple Hypothesis Tracking
MOP	Measure of Performance
MSDF	Multi Sensor Data Fusion
MTP	Maritime Tactical Picture
NRE	Non-Recurring Engineering
NTDS	Naval Tactical Data System
OGRO	Operator Guidance ReadOut
OGROA	Operator Guidance ReadOut Area
PADB	Performance Analysis Data Base

LIST OF ACRONYMS (cont'd)

PE	Performance Evaluation
QAB	Quick Action Button
R&D	Research and Development
RM	Resource Management
RMP	Recognized Maritime Picture
RTP	Real Time Performance
STA	Situation and Threat Assessment
TCC	Tactical Coordination Center
TIOI	Time Interval Of Interest
TSA	Tactical Situation Area
VOI	Volume Of Interest
2D	Two-Dimension
3D	Three-Dimension

1.0 INTRODUCTION

Multi-Source Data Fusion (MSDF) is about tactical picture compilation (Refs. 1-2). MSDF is concerned solely with individual entities (aircraft, missiles, etc.) considered in isolation (i.e., groups or formations of entities are not considered at this fusion level). An MSDF system processes the data and information reported by multiple dissimilar sources in order to correctly and quickly derive the best estimates of the current and future kinematics properties for each hypothesized (or perceived) entity in the operational environment, and to develop inferences as to the identity and key attributes of these entities.

The MSDF systems foreseen to support command teams onboard Canadian warships must be real-time systems. The correctness of the final implementation of a real-time system is not only a function of the accuracy of the computations performed by the system, but also a function of the time at which the results of the computations are produced. A real-time system monitors and controls some aspects of its environment. Typically, a real-time system waits for a stimulus from the environment. When the stimulus occurs, the computation associated with the response to that stimulus is started. The computation typically involves gathering some information about a specific aspect of the environment. The computation is completed when the output response associated with the stimulus is produced.

Responses may have timing constraints associated with them. A response must typically occur within some window of time from its triggering event sequence. A *hard timing constraint* is defined as a time window in which the response to the triggering event must occur for the system to be *correct*. The term *correct* may mean that the system will fail catastrophically if the deadlines are not met, with possible harm to people, property, or the environment. A *soft timing constraint* is defined as a window in which it is *desirable* for the response to be completed. This definition does not imply any penalty

if the constraint is not met. Consequently, it is of little value because it does not have any impact on the assessment of a system, either during acceptance testing or once deployed in the field.

Some prefer the notion of *quality of service* to the notion of hard and soft timing constraints. A real-time system that can meet all of its response time constraints may not be the best system possible because the responses may not be the best possible ones, but rather the best ones that could be produced within the available time and with the available resources. Traditional real-time system design tries to insure that a system can respond in a timely fashion to all events. This approach ignores the potential trade-off between timing behavior, accuracy of results, quantity of information processed and supplied, and cost. In an overload situation, a real-time system may respond to all events but degrade the quality (accuracy, quantity of information provided, etc.) of its responses, or it may respond in full to a subset of the events. Both approaches are termed *graceful degradation*. Should an MSDF system not be able to meet peak loading, it might drop some input data sets or it may reduce the quality of the tracks and identity propositions that are maintained. Other trade-off may be possible.

The notion of *quality of service* is a generalization of the notion of timing constraints, graceful degradation, and requirement. It embodies the idea that not only must the system behave in a timely manner that can be mathematically specified, but it must also produce acceptable responses. If a fusion algorithm were to quickly but erroneously identify a friendly track as hostile, its quality of service would be considered low because the friendly may be prosecuted. Similarly, its quality of service would be low if it identified a hostile track correctly but too late, e.g., after the hostile had engaged the ship.

Some real-time systems only collect information about the environment; they do not interact with it. Typically, MSDF systems are such systems. However, an MSDF system that is used onboard a warship fits within a larger system that would normally set

up a feedback loop. The crews on the Halifax class ships would use the information provided by the MSDF system to modify the environment of the system. They could change the mode of operation of the sensors to gather more information, they could change the course and speed of the ship, or they could engage some targets.

From a Research and Development (R&D) perspective, MSDF system developers need capabilities that allow them to assess if a proposed MSDF System is working properly, as expected. In view of the above discussion, there are two major aspects that must be taken into account to perform this assessment:

1) the performance of the MSDF System as a target tracking and identification process (i.e., does the MSDF System compute the "correct", "expected" output given the stimulation data provided?), and

2) the performance of the MSDF System as a real-time system (i.e., is the MSDF output provided in a timely manner with respect to the real-time constraints specified for the system?).

A set of tools is thus required to support the designer/developer/user/operator in quantitatively assessing the performance of the MSDF algorithms and techniques. It was identified early in the MSDF project at DREV that rigorously evaluating candidate MSDF algorithms using real sensor and link data with meaningful measures of performance would not be practical. For example, real data do not include ground truth information, thereby reducing the possibilities of comparing the *perceived* situation with the *real* situation in the environment. One also has to face the extremely limited availability of trial data to support algorithm analysis and development and MSDF system prototyping. Many research programs cannot afford to incur the additional costs of data collection for the purpose of demonstrating concepts with real data. Alternatives to this situation include artificially synthesizing appropriate data from trial data collected under non-standard conditions (not easy to do in a convincing manner), or to employ sufficiently

high-fidelity simulators. This last option has been retained for the R&D activities aiming at exploring the MSDF concepts at DREV since, most of the time, representative simulated data may be adequate to verify or validate MSDF concepts.

Consequently, a highly modular, structured, and flexible simulation environment has been developed as a proof-of-concept demonstrator for MSDF. This simulation environment (Refs. 3-5) is called CASE_ATTI (Concept Analysis and Simulation Environment for Automatic Target Tracking and Identification). For CASE_ATTI, a high-fidelity simulator that emulates the behavior of real targets, sensor systems of the HALIFAX class ships and the meteorological environment has been developed to generate simulated sensor and link data to stimulate the MSDF systems under evaluation. It allows the user to create and edit test scenarios with multiple ships/sensors/targets. The ships can be stationary or moving along predefined paths. One or several sensors can be assigned to each ship. Targets are created with defined trajectories and attributes.

In addition to high-fidelity stimulation, one of the main requirements of the CASE_ATTI test bed has been to provide the algorithm-level test and replacement capability required to study and compare the applicability and performance of advanced, state-of-the-art MSDF techniques.

Unfortunately, even though CASE_ATTI is a very powerful tool, it does not provide all of the capabilities required to meet all of the performance assessment needs for MSDF systems. CASE_ATTI is a non-real-time proof-of-concept simulation environment. Therefore, albeit past work with MSDF systems clearly indicates that these future naval systems, built using a combination of numerical and artificial intelligence technologies, will have severe data and information processing requirements placed upon them, the implementation of a real-time MSDF function taking into account the constraints imposed by the complete Command and Control (C2) system is a critical aspect of the MSDF domain which has only been given limited consideration so far.

A separate means is thus required to investigate the real-time aspects of the MSDF systems proposed for the future. This document gives a detailed description of such a means, hereafter called the "MSDF Test Bed", that will help to establish the real-time requirements of MSDF systems on speed, responsiveness, predictability, timeliness, etc. At the same time, the MSDF Test Bed will also provide a capability to evaluate various measures of performance for these MSDF Systems.

Chapter 2.0 first provides some useful definitions of the basic terminology used in this document and then gives an overview description of the MSDF Test Bed in terms of the main interacting elements (i.e., the modules) that compose the system. The purpose of each module is described along with how these modules work and interact together to achieve the overall functionality and objectives of the MSDF Test Bed. However, not all of these modules are used simultaneously during the execution of the MSDF Test Bed. Different operating conditions involve different modules. Hence, Chap. 3.0 provides an overview of the MSDF Test Bed in terms of system states and operating modes that complements the decomposition into modules and components.

Then, Chaps. 4.0 to 7.0 give detailed descriptions of the Stimulation, Performance Analysis Database (PADB), Human Computer Interface (HCI) and Performance Evaluation (PE) modules, respectively. The main conclusions are summarized in Chap. 8.0, while references are provided in Chap. 9.0.

This work was carried out at DREV between September 1995 and May 1996 under Work Units 1ba12 (Investigations of MDSF/STA/RM Concepts) and 1ba14 (Support to the ASCACT Integration Working Group).

2.0 MSDF TEST BED SYSTEM OVERVIEW

This chapter provides an overview description of the MSDF Test Bed in terms of the main interacting elements that compose the system. Figure 1 shows the MSDF Test Bed block diagram.

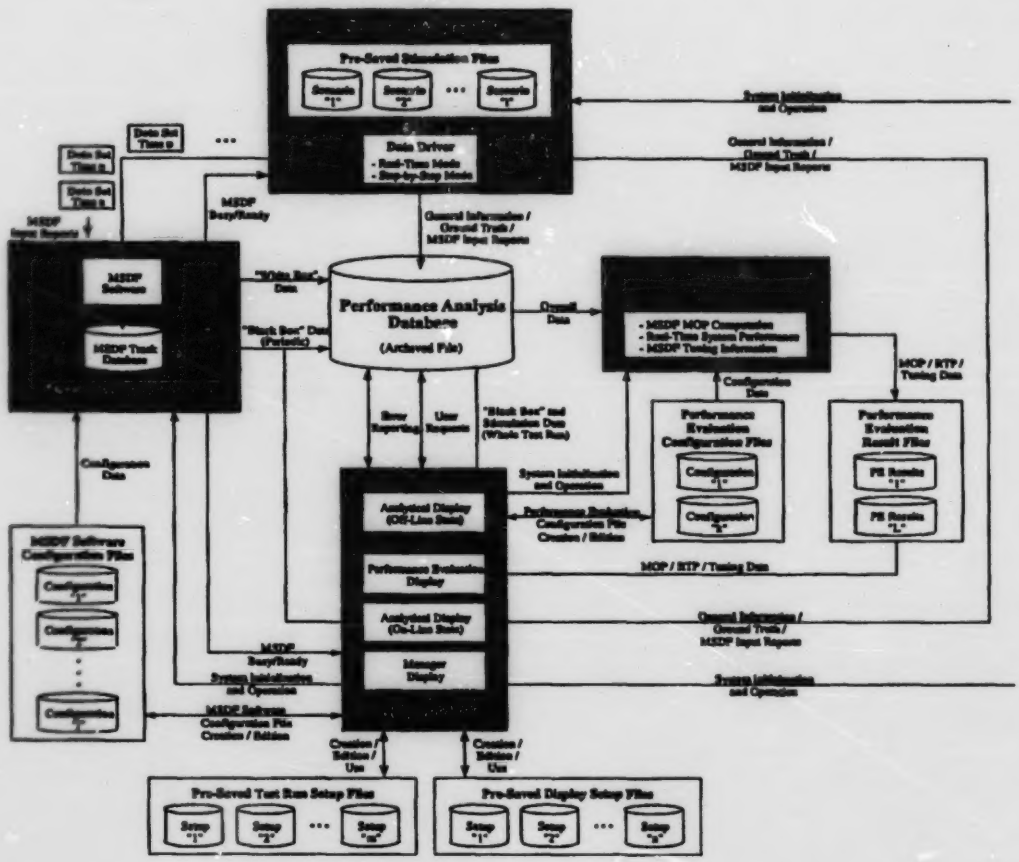

FIGURE 1 - The MSDF Test Bed

The main elements of the MSDF Test Bed are the Stimulation module, the MSDF System module, the Human Computer Interface (HCI) module, the Performance Evaluation (PE) module and the Performance Analysis Database (PADB). These modules are briefly described below, following some useful definitions of the basic terminology used in this document.

The software composing the MSDF Test Bed must be developed with reusability in mind. Minimal non-recurring engineering (NRE) should be required to add new measures of performance (MOPs), to modify the display features, etc.

2.1 Definitions

For convenience, some brief definitions of the basic terminology used in this document are provided in this section. The MSDF Test Bed is a large and complex distributed real-time system. Its decomposition in terms of modules, components, states, modes, functions and capabilities accounts for this complexity and facilitates the understanding of the various concepts. A hierarchy has been established such that the modules are decomposed into components and the states are decomposed into modes that are themselves broken into functions and capabilities.

2.1.1 Module

A module is defined in this document as some software that is organized into limited aggregates of data and contiguous code that perform identifiable operations associated with a specific concept of the overall system (e.g., MSDF System module, Stimulation module, etc.). Each module is given specific responsibilities that may require communication with other modules.

The overall goal of the decomposition of a system into modules is the reduction of software cost by allowing modules to be designed and revised independently. Each module's structure should be simple enough that it can be understood fully. It should be

possible to change the implementation of a module without knowledge of the implementation of other modules and without affecting the behavior of other modules. The ease of making a change in the design should bear a reasonable relationship to the likelihood of the change being needed.

2.1.2 Component

The code for a module may still be of considerable size. Hence, modules are therefore subdivided as necessary into components. A computer software component is thus a specialized, functional or logically distinct, part of a module (e.g., MSDF Software component, MSDF Track Database component, etc.).

2.1.3 State

Typically, the decomposition into modules and components as discussed above does not emphasize the dynamic aspects of the system. Potentially, not all of the modules are used simultaneously during the execution of a program. Typically, different operating conditions involve different modules. Hence, the system can also be decomposed in terms of system states and operating modes (e.g., using a state transition diagram) that complements the decomposition into modules and components.

A "state" is thus defined as a condition in which the system can exist (e.g., control state, on-line state, off-line state).

2.1.4 Mode

A "mode" is defined as a particular functioning arrangement of the state operations (e.g. real-time mode, replay mode, etc.). Each mode has its specific, distinct purpose and as such, it involves only the subset of the modules that is required to achieve the objective of the mode.

2.1.5 Function

A function is defined as a group of related capabilities of a mode. In a sense, it is used to decompose a mode into submodes.

2.1.6 Capability

A capability is defined as one or a group of related requirements of a mode and/or function. It defines the most basic level of functionality related to a single practical concept related to the system implementation (e.g., file selection, test run launching and abortion, error reporting, etc.).

2.2 Development Environment

The Development Environment of the MSDF Test Bed includes software (e.g., network simulation and monitoring software, compiler, relational database management system, word processing software, etc.), and hardware.

2.3 Stimulation Module

The Stimulation module must provide realistic sensor and link data to the MSDF System module in order to allow the end user to evaluate the performance of this MSDF System in representative conditions. However, the use of real sensor data is impractical for the MSDF Test Bed. Hence, high-fidelity sensor and link simulations will be used instead. Since the required level of fidelity for these simulations could be hard to achieve in real time, the MSDF Test Bed will use pre-saved sensor and link data files (called stimulation files) generated using an external high-fidelity simulation environment for the scenarios of interest. The sensor and link data archived in these data files must be time stamped so that the Data Driver component of the Stimulation module can read the files and control the transfer of their contents to the MSDF System module (and to the other modules) at the appropriate time.

2.4 MSDF System Module

The MSDF System module is composed of two main components, i.e., the MSDF Software and the MSDF Track Database. The MSDF Software receives its input data (i.e., sensor and link data) from the Stimulator module.

The data processing performed by the MSDF Software is divided into a number of blocks such as target state estimation and kinematic behavior assessment, track management (i.e., track initiation, confirmation, deletion, etc.), cluster management (i.e., cluster merging, splitting, etc.), contact-to-track association and track-to-track association, track fusion, target identity information fusion, sensor data alignment (both spatial and time alignment), and internal track database. A detailed description of these tracking and identification algorithms and processes is provided in Ref. 2.

During its operation, the MSDF Software component generates an estimated tactical picture that should accurately reproduce the ground truth tactical picture. This perception of truth by the MSDF Software is embodied in the tracks that are established and maintained as the sensors sample the environment. Hence, the MSDF Track Database contains the resulting track data obtained after each new sensor or link report has been processed and fused (i.e., the periodic updates of the kinematic and identity information maintained for each track). These data, also called "black box" data in this document (see Appendix A), must be sent to both the PADB and the HCI modules. Typically, each track record in the MSDF Track Database includes a state vector (with a corresponding covariance matrix) estimating the target kinematic properties, and one or more propositions about non-kinematic properties of the target, each with its associated likelihood function. The information kept in a track record also includes a time tag corresponding to the last update time, the quality of the track, the blip/scan parameter for each sensor, etc. The exact format for the MSDF Track Database can only be formulated for a specific application.

Upon user request, the MSDF Software component must also output some intermediate results of interest produced by the MSDF algorithms while they are running. These data are called "white box" data in this document (see Appendix A). They describe the actual processing that has been performed by the MSDF Software (i.e., its internal behavior) to transform the input sensor and link data into the resulting MSDF tracks at the output. These data are mostly required to perform detailed performance evaluation analyses during the development and tuning of the MSDF Software. Clearly, the real-time performance of the MSDF Software can be affected when some parameter values are probed inside the MSDF System while it is running. Hence, this data must be extracted only when strictly necessary, and the MSDF output results must then be analyzed with great care. The exact format for the MSDF white box data can only be formulated for a specific application.

The MSDF System module must also output to the Stimulation and HCI modules the MSDF Busy/Ready signals reflecting the processing status of the MSDF Software. When appropriate, these signals are used by the end user or the Data Driver component to control the transfer of the stimulation file contents to the MSDF System module (and to the other modules) at the appropriate time.

The MSDF System module must accept system initialization (e.g., MSDF configuration file name, etc.) and operation (e.g., run, stop, etc.) data/commands from the HCI module. The MSDF System module must also accept the MSDF Software configuration data that allow the various MSDF algorithms and processes to configure themselves prior to any test run. These configuration data could include such things as the maximum number of hypotheses to be kept by an MHT (Multiple Hypothesis Tracking) algorithm for each input data set cycle, the assumed values of the target model process noise to be used by an adaptive Kalman filter, etc.

2.5 Performance Analysis Database (PADB)

The Performance Analysis Database (PADB) contains an archive of all the data manipulated by the MSDF Test Bed. The archived data for a given test run include the white and black box data produced by the MSDF System, the stimulation data provided by the Data Driver component, the error reporting data generated by the MSDF Test Bed and the log data about any user requests that may affect the MSDF output results.

2.6 Performance Evaluation Module

The main purpose of the Performance Evaluation (PE) module is to provide a capability that allows the end user to assess if the MSDF System is working properly, as expected. There are two major aspects that must be taken into account to perform this assessment:

1) the performance of the MSDF System as a target tracking and identification process (i.e., does the MSDF System compute the "correct", "expected" output given the stimulation data provided?), and
2) the performance of the MSDF System as a real-time system (i.e., is the MSDF output provided in a timely manner with respect to the real-time constraints specified for the system?).

The PE module thus comprises a set of tools to support the designer/user/operator in his/her quantitative assessment of the performance of the MSDF algorithms and techniques. These tools include a compilation mechanism for the tracking statistics, computation mechanisms for the various measures of performance, etc.

2.7 Human Computer Interface

This module provides all of the Human Computer Interface (HCI) facilities required to support all interactions of the MSDF Test Bed with its user/operator. Indeed,

the HCI module is the top-level module of the MSDF Test Bed, and as such, it manages (i.e., integrate and control) all other modules.

Through the Manager Display component, the HCI module must allow user interactions with the other modules (e.g., by prompting for values of specific configurable internal parameters for the other modules). Through the Analytical Display and PE Display components, the HCI module provides the capability to present the stimulation scenario data, the input/output data of the MSDF System, the MSDF Test Bed error/alarm data, and the results of the performance analyses of the MSDF System.

The HCI module must be designed to be intuitive, easy to use (i.e., it must be ergonomic) and modifiable. The HCI module has to use modern windowing (and other interfacing) capabilities.

3.0 MSDF TEST BED STATES AND MODES

The brief description given in the previous chapter provides a general understanding of the MSDF Test Bed. The purpose of each module is described along with how these modules work and interact together to achieve the overall functionalities and objectives of the MSDF Test Bed. However, not all of theses modules are used simultaneously during the execution of the MSDF Test Bed. Different operating conditions involve different modules.

This chapter provides an overview of the MSDF Test Bed in terms of system states and operating modes that complements the decomposition into modules and components presented above.

3.1 States/Modes Transitions

The MSDF Test Bed exists in three (3) states. These states are as follows:

i. Control State
ii. On-Line State
iii. Off-Line State

Figure 2 shows the relationship between the MSDF Test Bed states and their associated modes. A brief description of each of these states and their corresponding modes is given in the following sections. The two main states are the On-Line and the Off-Line states. The basic distinction between these two states lies in both the status of the MSDF System module and the performance evaluation activities performed while the MSDF Test Bed is up and running.

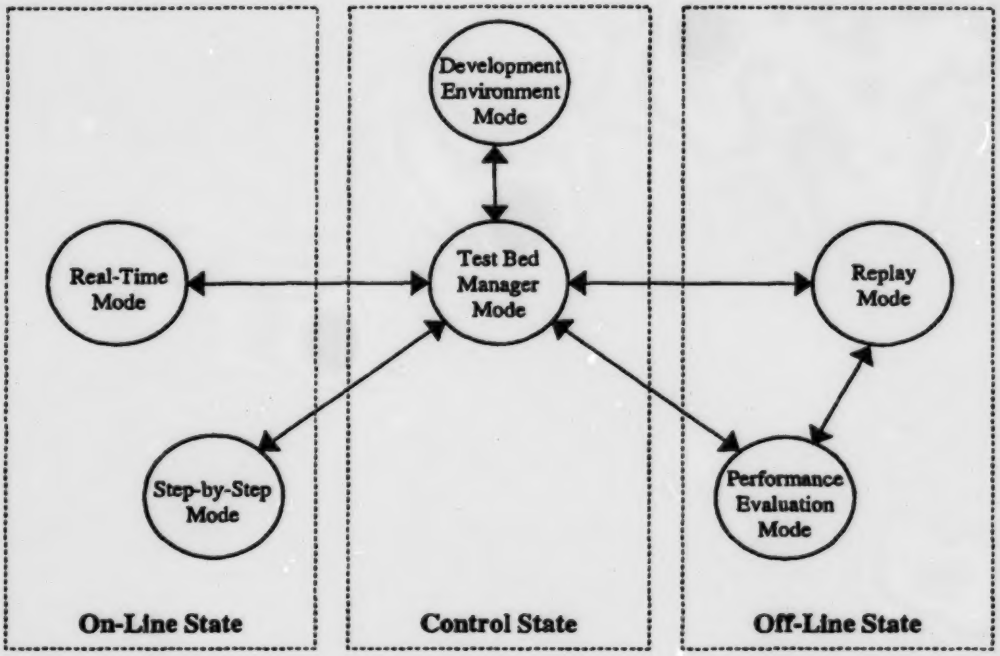

FIGURE 2 - The MSDF Test Bed states and modes transitions

3.2 Control State

The Control state is the default initialization state of the MSDF Test Bed. This state has two modes of operation:

i. Development Environment mode
ii. Test Bed Manager mode

A description of each of these modes is given in the following sections.

3.3 Control State / Development Environment Mode

In the Development Environment mode, the MSDF Test Bed user has access to all operating system functions to perform the required housekeeping activities and system

management activities (e.g., data file backup, file copying, file renaming, etc.). The user also has access to all of the development environment tools (compilers, etc).

The user may also transition to the Test Bed Manager mode, thereby launching the MSDF Test Bed.

3.4 Control State / Test Bed Manager Mode

In the Test Bed Manager mode, the MSDF Test Bed must coordinate and manage all mode transitions as illustrated in Fig. 2. Hence, this mode must provide the user with a capability to access/transition to one of the following modes of the MSDF Test Bed:

- Real-Time
- Step-by-Step
- Replay
- Performance Evaluation
- Development Environment

The Test Bed Manager Mode must also provide the user with a capability to terminate the MSDF Test Bed. Upon program termination, the MSDF Test Bed must return to the Development Environment mode of the Control state.

A mechanism must be implemented to handle any abnormal termination event that may arise during the program execution. When the MSDF Test Bed detects an abnormal termination condition, it must kill all of the running processes and close all of the opened files. Upon abnormal program termination, the MSDF Test Bed must return (if possible) to the Development Environment mode of the Control state.

3.5 On-Line State

In the On-Line state, the sensor and link data provided by the Stimulation module are processed live, "on-line" by the MSDF System. The stimulation data and the resulting

MSDF tracks must then be displayed "on-line" (i.e., as they are generated) by the HCI module. Finally, all of these data must be archived "on-line" in the PADB. In this state, the Analytical Display component allows for a quick assessment of the MSDF System performance. However, no extensive performance evaluation activities (other than the strictly necessary data collections) are performed since such activities could greatly affect the real-time performance of the MSDF Software if performed while it is running.

The On-Line state has two (2) corresponding modes. These modes are as follows:

i. Real-Time mode
ii. Step-by-Step mode

The basic distinction between these two modes lies in the data transfer conditions applied by the Data Driver component of the Stimulation module.

3.6 On-Line State / Real-Time Mode

When the Data Driver is operating in the Real-Time mode, it controls the transfer of the stimulation file contents so that each data set is transferred only when the time stamp written in that data set corresponds exactly to the current simulation time (i.e., the rate is adjusted to perfectly match the time stamp of the data sets with the current simulation time). The current simulation time is the actual time since the beginning of the simulation. Hence, in the Real-Time mode, the MSDF System really operates as a real-time system. Since the main purpose of the MSDF Test Bed is to establish the real-time requirements of the MSDF System, this is the most fundamental mode of the testbed.

In this mode, data flow occurs as depicted in Fig. 3. The data manipulated by the MSDF Test Bed must then be verified, displayed and archived as per the user settings.

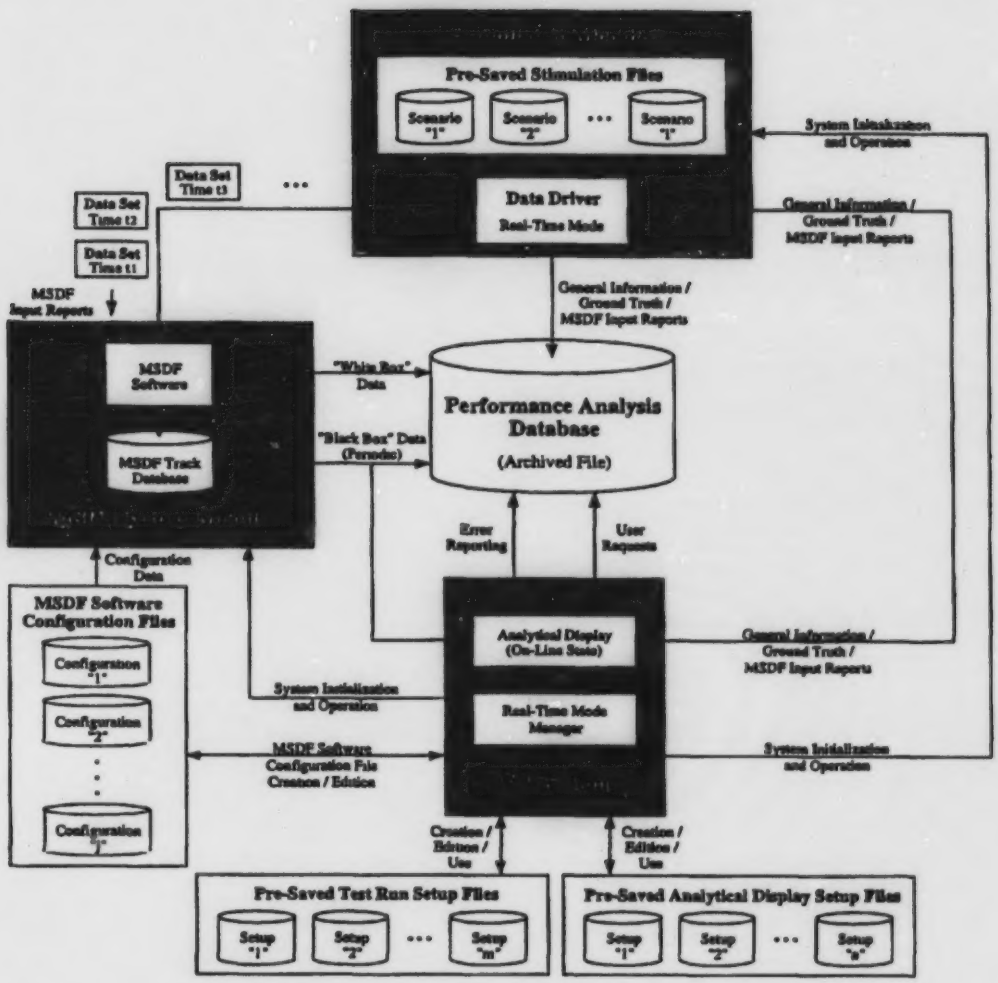

FIGURE 3 - On-Line State / Real-Time Mode modules

3.7 On-Line State / Step-by-Step Mode

When the Data Driver component is operating in the Step-by-Step mode, it waits until it receives a data request before it sends the next block of data sets that all have the same time stamp. While the MSDF System module is processing a given block of data

sets (all having the same time stamp), it outputs the MSDF Busy signal. As soon as the MSDF System completes the processing of this block of data sets, it sends an MSDF Ready signal. Upon reception of this MSDF Ready signal, the data request can be initiated manually by the user, or automatically by the MSDF Test Bed. Upon such a request, the Data Driver accordingly sends the next block of data sets.

This mode ensures that the MSDF System has sufficient time to process all input reports. It takes advantage from the fact that the MSDF Test Bed is using pre-saved sensor and link data instead of real-time stimulation data. Hence, there is an opportunity with this mode to dissociate the algorithmic nature of the MSDF System from its real-time performance. In other words, this mode can be used to check if the MSDF System results are affected by the real-time constraints (i.e., are the tracking and identification results different when the MSDF System has plenty of time to process the input data?).

In the Step-by-Step mode, data flow occurs as depicted in Fig. 4. The data manipulated by the MSDF Test Bed must then be verified, displayed and archived as per the user settings.

The user of this mode has the possibility to select one out of three functions that affect the behavior of the Stimulation module (and, as a consequence, the other modules as well). These functions are:

i. Manual: The user must request the next block of data sets from the Data Driver component.
ii. Automatic: The MSDF System, when ready to accept new data, must request the next block of data sets from the Data Driver component.
iii. Real-Time Like: The MSDF System receives the data sets from the Data Driver component as in the Real-Time mode.

and they are further discussed in Chap. 4.0.

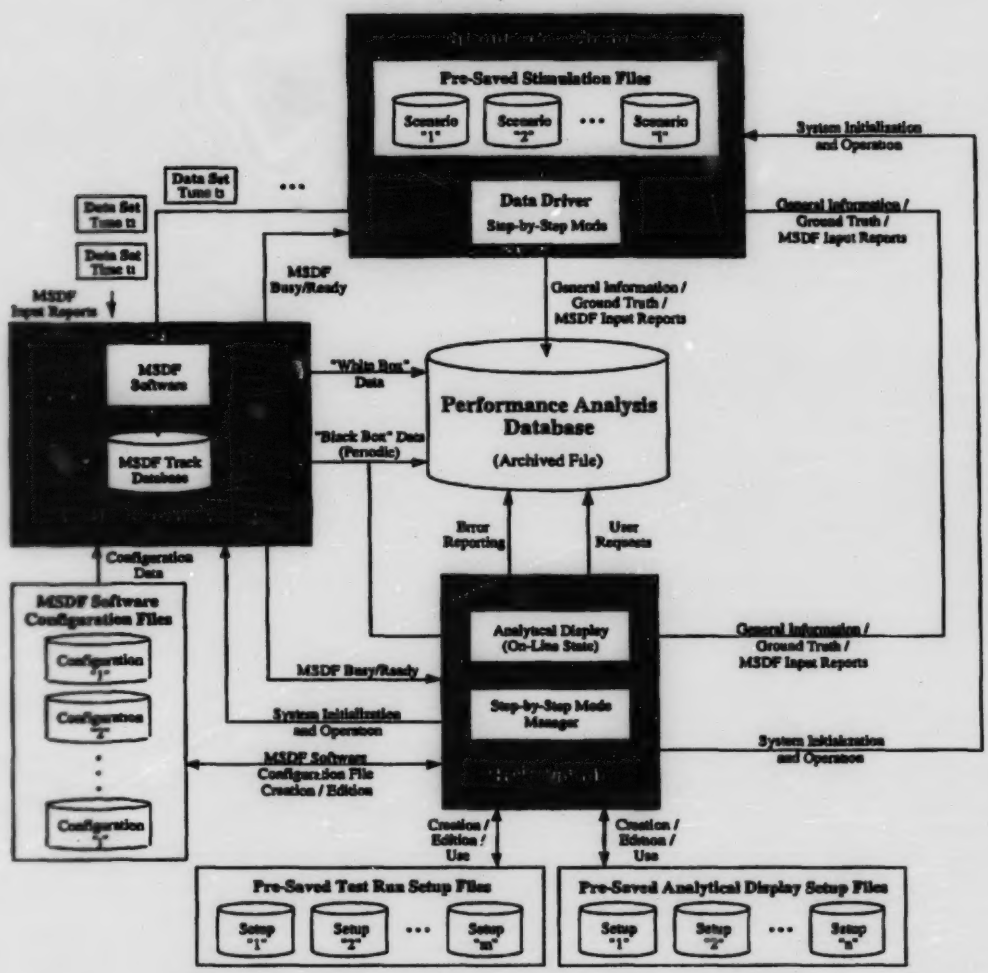

FIGURE 4 - On-Line State / Step-by-Step Mode modules

3.8 Off-Line State

In the Off-Line state, the Stimulation and MSDF System modules are not operational. The data processed and displayed by the PE and the HCI modules must be retrieved from an archive file of the PADB that contains all the archived data for a test run previously performed in the On-Line state. The main purpose of this state is to

conduct detailed MSDF performance evaluation analyses "off-line", without having to care about disturbing the real-time processing activities of the MSDF System. This state must also allow the user to simply replay a test run for visualization purpose (e.g., to more closely investigate some interesting target tracking phenomenon that happens at some specific moment of the scenario) without having to actually run the Stimulation and MSDF System modules.

The Off-Line state has two (2) corresponding modes. These modes are as follows:

i. Replay mode
ii. Performance Evaluation (PE) mode

3.9 Off-Line State / Replay Mode

In the Replay mode, the user has the capability to replay a test run for visualization purposes (e.g., to more closely monitor some interesting target tracking phenomenon that happens at some specific moment of the scenario) without having to actually run the Stimulation and MSDF System modules. All stimulation and black box data contained in a time slice (i.e., a time window) specified by the user must be retrieved from the PADB and be presented simultaneously on the Analytical Display. Since all of the pertinent data for the test run have already been computed and archived, the user of this mode has plenty of time to investigate any issues of interest and to focus on specific aspects that require special consideration.

In the Replay mode, data flow occurs as depicted in Fig. 5. The data manipulated by the MSDF Test Bed must then be verified and displayed as per the user settings.

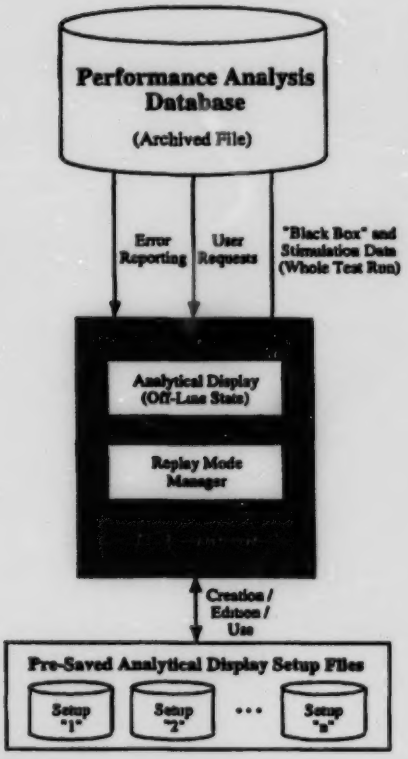

FIGURE 5 - Off-Line State / Replay Mode modules

3.10 Off-Line State / Performance Evaluation Mode

The main purpose of the Performance Evaluation (PE) mode is to conduct the detailed analyses required to quantitatively assess the target tracking and identification performance of the MSDF System and its real-time performance as a real-time system. The PE mode must also provide a capability to analyze the archived data for the purpose of tuning the MSDF algorithms.

In the PE mode, data flow occurs as depicted in Fig. 6. The data manipulated by the MSDF Test Bed must be verified and displayed as per the user settings.

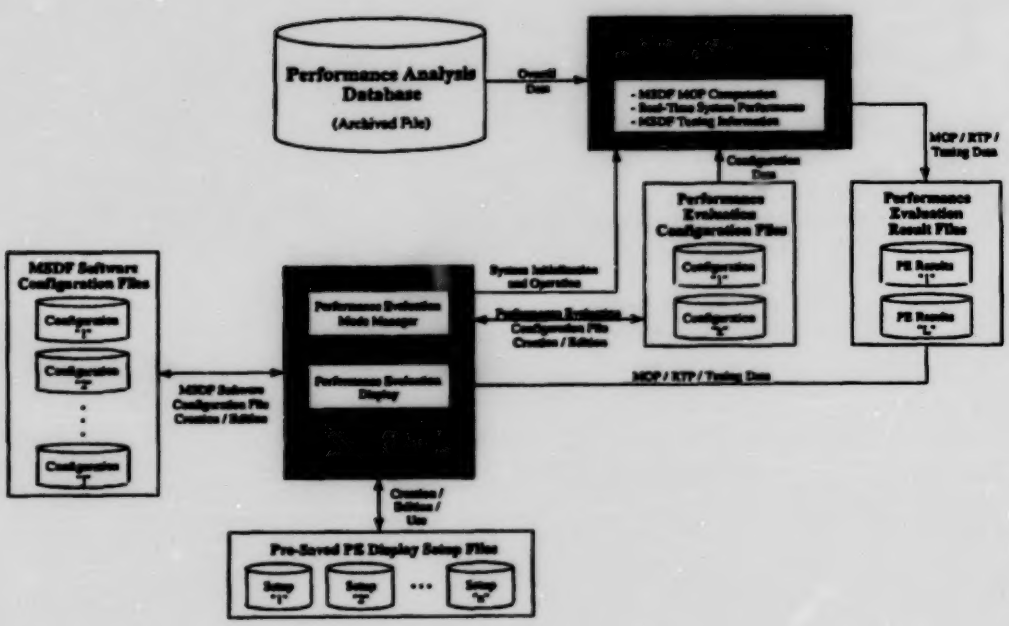

FIGURE 6 - Off-Line State / Performance Evaluation Mode modules

Since the visualization of the stimulation and MSDF data and the computation of the various measures of performance are two activities of the analysis process that are highly related, the user must have the capability to quickly switch from the Replay mode to the PE mode, and vice versa. Indeed, it is highly desirable that a direct link exist between the Analytical Display and the PE Display so that, for example, a modification of the time interval of interest on the Analytical Display automatically produces the corresponding modification on the PE Display, and vice versa.

4.0 STIMULATION MODULE

In the On-Line state, the Stimulation module must provide realistic sensor and link data to the MSDF System module in order to allow the end user to evaluate the performance of this MSDF System in representative conditions. The MSDF Test Bed uses pre-saved sensor and link data files (called stimulation files) generated using an external high-fidelity simulation environment for the scenarios of interest to MSDF. The sensor and link data archived in these data files must be time stamped so that the Data Driver component of the Stimulation module can read the files and control the transfer of their contents to the MSDF System module (and to the other modules) at the appropriate time.

4.1 Stimulation Module Inputs / Outputs

The Stimulation module must accept system initialization (e.g., mode specification, Step-by-Step mode function specification, stimulation file name, etc.) and operation (e.g., run, stop, request for the next data set, etc.) data/commands from the HCI module. In the Step-by-Step mode, the Stimulation module must also accept the MSDF Busy/Ready signals from the MSDF System module. These status signals are used by the Data Driver component to control the transfer of the stimulation file contents to the MSDF System module (and to the other modules) at the appropriate time.

4.2 Pre-Saved Stimulation Files

The simulated sensor and link data generated for a given MSDF scenario of interest must be stored in the stimulation file as a sequence of time stamped data sets. The information should be stored in ASCII. The types of data sets that can typically be found in the stimulation file are:

- General Information Data Set
- Radar Data Set

- IFF Data Set
- ESM Data Set
- Link Data Set
- Navigation Data Set
- Ground Truth Data Set

Except for the General Information data set that must appear only once and is always the first data set, the exact number and sequence of each data set type in the stimulation file is scenario dependent.

Reference 6 discusses in detail one possibility for the generation of the simulated sensor and link data required for the MSDF Test Bed.

4.3 Data Driver

The Data Driver component reads a pre-saved stimulation file and must control the transfer of its contents to the MSDF System, HCI (Analytical Display) and PADB modules of the MSDF Test Bed. The simulated sensor and link data stored in a stimulation file can typically be decomposed as:

- General Information
- MSDF Input Reports
  - Contact Data
  - Track Data
  - Navigation Data
- Ground Truth Data

During the execution of the MSDF Test Bed, only a subset of these data (i.e., the MSDF input reports) is sent to the MSDF System module, while all of the data must be sent to the HCI module and the Performance Analysis Database. Hence, the Data Driver component of the Stimulation module must perform some filtering of the data provided in

the pre-saved stimulation file. The Data Driver component must also perform the routing of these data to the appropriate modules of the MSDF Test Bed.

4.3.1 Data Driver / Real-Time Mode

When the Data Driver is operating in the Real-Time mode, it controls the transfer of the stimulation file contents so that each data set is transferred only when the time stamp written in that data set corresponds exactly to the current simulation time (i.e., the rate is adjusted to perfectly match the time stamp of the data sets with the current simulation time). The current simulation time is the actual time since the beginning of the simulation.

4.3.2 Data Driver / Step-by-Step Mode

When the Data Driver component is operating in the Step-by-Step mode, the Data Driver waits until it receives a data request before it sends the next block of data sets that all have the same time stamp. The data request can be initiated manually by the MSDF Test Bed user, or automatically by the MSDF Test Bed. Within the Step-by-Step mode, a Real-Time Like function must also be available, allowing portions of the test run to be performed in real time. The Manual Step-by-Step function is the default function automatically selected when a test run is launched in the Step-by-Step mode.

This mode ensures that the MSDF System has sufficient time to process all input reports.

4.3.2.1 Manual Step-by-Step Function

The Manual function of the Step-by-Step mode is the function where the Data Driver component receives its data request from the user via the Step-by-Step Mode Manager component of the HCI module. While the MSDF System is processing a block of data sets (all having the same time stamp), the HCI must display the MSDF Busy signal. As soon as the MSDF System completes the processing of this block of data sets,

it must send an MSDF Ready signal that must be displayed to the user. The user then has the possibility of initiating the next data request to the Data Driver. Upon such a request, the Data Driver must accordingly send the next block of data sets.

When the MSDF Ready signal is received and displayed, the user must have the possibility of switching to the Automatic Step-by-Step function or to the Real-Time Like Step-by-Step function instead of sending a manual data request. Before switching to a different function, the user has to specify a simulation time period during which the Data Driver will remain in that new specified function. While a portion or all of the test run is being performed in the new specified function, the user must have the possibility, through a synchronous user input, of switching the new specified function back to the Manual Step-by-Step function even if the requested simulation time period has not been completed.

4.3.2.2 Automatic Step-by-Step Function

The Automatic function of the Step-by-Step mode is the function in which the Data Driver component receives its data request directly from the MSDF System. As soon as the MSDF System completes the processing of a block of data sets (all having the same time stamp), it must send a data request (i.e., the MSDF Ready signal) directly to the Data Driver.

Before switching to this function, the user must specify the simulation time period during which the Data Driver has to output the blocks of data sets under the control of the MSDF System (i.e., as requested by the MSDF System). While a portion or all of the test run is being performed under the Automatic Step-by-Step function, the user has the possibility, through a synchronous user input, of switching back to the Manual Step-by-Step function even if the requested simulation time period has not been completed.

4.3.2.3 Real-Time Like Step-by-Step Function

The Real-Time Like function of the Step-by-Step mode must allow a portion or all of the test run to be performed as if it were in real-time. The actual time interval between each transfer of a block of data sets must then be adjusted to perfectly match the time interval corresponding to the time stamp difference of the involved data sets.

Before switching to this function, the user must specify the simulation time period during which the Data Driver has to output the blocks of data sets in real-time. While a portion or all of the test run is being performed under the Real-Time Like Step-by-Step function, the user has the possibility, through a synchronous user input, of switching back to the Manual Step-by-Step function even if the requested simulation time period has not been completed.

5.0 PERFORMANCE ANALYSIS DATABASE (PADB)

This chapter discusses the Performance Analysis Database (PADB) that contains archives of the data manipulated by the MSDF Test Bed.

5.1 PADB in the On-Line State

In the On-Line state, the HCI module and the MSDF Software, MSDF Track Database and Data Driver components must archive to the PADB. The archived data for a given test run include the white and black box data produced by the MSDF System, the stimulation data provided by the Data Driver component, the error reporting data generated by the MSDF Test Bed and the log data about any user requests that may affect the MSDF output results. All of these data are necessary in the Off-Line State of the MSDF Test Bed to perform the replay and performance evaluation activities.

As mentioned in Subsection 4.3.2, the user of the Step-by-Step mode has the possibility of selecting one out of three functions that affect the behavior of the Stimulation module (and, as a consequence, the other modules as well). These functions are Manual, Automatic and Real-Time Like. The user requested transitions from one Step-by-Step function to the other must be logged into the PADB since the output of the MSDF System must be reproducible and these transitions may also affect the MSDF results for the test run. With respect to the PADB, this is the only difference between the Real-Time mode and the Step-by-Step mode.

If necessary (i.e., as requested by the user), the MSDF Test Bed must be capable of performing the data archiving function over the entire duration of a test run. The archiving capability may also be inactive as a result of the Data Archiving Status Control capability described in Subsection 6.2.6.3.

When operating in the On-Line state of the MSDF Test Bed, the modules must have write-only access to the PADB.

5.1.1 Stimulation Data Logging

The Stimulation module must archive to the PADB all of its output data. Reference 6 provides representative formats for the various data sets that have to be transferred to the PADB by the Data Driver component.

5.1.2 Black Box Data Logging

The MSDF Track Database component must archive all of the MSDF track data to the PADB. These data are also called "black box" data in this document. Typically, each track record in the MSDF Track Database includes a state vector (with a corresponding covariance matrix) estimating the target kinematic properties, and one or more propositions about non-kinematic properties of the target, each with its associated likelihood function. The information kept in a track record also includes a time tag corresponding to the last update time, the quality of the track, the blip/scan parameter for each sensor, etc. Appendix A provides a representative format for the track records that must be archived to the PADB.

5.1.3 White Box Data Logging

Upon user request, the MSDF Software component must archive to the PADB some intermediate results of interest produced by the MSDF algorithms while they are running. These data are called "white box" data in this document. They describe the actual processing that has been performed by the MSDF Software (i.e., its internal behavior) to transform the input sensor and link data into the resulting MSDF tracks at the output. Appendix A provides an example of such white box data that must be archived to the PADB.

5.1.4 Error Reporting Data Logging

Through the HCI module, the MSDF Test Bed must accomplish error reporting and must log the appropriate error event descriptions to the PADB. The error reporting

data must include any appropriate data that describe the anomalies that occurred while the MSDF Test Bed was up and running. The error description data must be sufficiently detailed to allow the designer/user/operator to understand the problem and fix it if necessary. This data must also include a description of the minor errors that do not cause the system to crash, but that may affect the MSDF output results.

As the bare minimum, at least the following events must be archived:

i. Receipt of invalid data from the Data Driver (e.g., a corrupted data set being sent to the MSDF System module).
ii. MSDF Software input FIFO buffer overflow.
iii. Analytical Display FIFO buffer overflow.
iv. PADB FIFO buffer overflow.

In the case of item (iv) above, a mechanism must be put in place (i.e., an auxiliary file) to ensure that this error event is properly logged to the PADB and is not lost due to the overflow of the PADB buffer. This mechanism must ensure the integrity of the error log file.

5.1.5 User Requests Data Logging

Through the HCI module, the MSDF Test Bed must trap and embed within the archive file all those user requests (such as "start archiving white box data", "switch to the Real-Time Like function for some time period", etc.) that may have an effect on the MSDF output results. The archived data must also include all commands and events which affect the data recorded. Examples of such commands are:

- Stop Archiving
- Start Archiving

5.1.6 History of Step-by-Step Function Transitions Logging

The output of the MSDF System must be reproducible. However, using the Real-Time Like function of the Step-by-Step mode affects the processing time available to the MSDF System, which in turn may affect its output. In order to be able to reproduce the exact same MSDF System output for a given stimulation file, a log of the history of the transitions from one Step-by-Step mode function to the other must be produced. This log of the Step-by-Step mode function history must be easily accessible. The user must have the possibility to reproduce the exact same MSDF output, during a subsequent Step-by-Step test run, simply by entering the sequence of Automatic and Real-Time Like function transition requests (with their associated simulation time periods) as indicated on the log.

5.2 PADB in the Off-Line State / Replay Mode

In the Replay mode, the HCI module must retrieve data from the PADB. The retrieved data for a given test run include the MSDF track data (i.e., the black box data) produced and archived by the MSDF System and the stimulation data transferred and archived by the Data Driver component. The error reporting data generated and archived by the MSDF Test Bed and the log data about any user requests that may affect the MSDF output results must also be retrieved since these data might prevent the HCI module to manipulate corrupted data.

In this mode of operation of the MSDF Test Bed, the modules have read-only access to the PADB.

5.2.1 Stimulation Data Retrieving

In the Replay mode, the HCI module must retrieve from the PADB the stimulation data previously transferred and archived by the Data Driver component.

5.2.2 Black Box Data Retrieving

In the Replay mode, the HCI module must retrieve from the PADB the MSDF track data previously transferred and archived by the MSDF Track Database component.

5.2.3 Error Reporting Data Retrieving

In the Replay mode, the HCI module must retrieve from the PADB the error reporting data previously generated and archived by the MSDF Test Bed. These data are used by the HCI module to prevent the manipulation of corrupted data.

5.2.4 User Requests Data Retrieving

In the Replay mode, the HCI module must retrieve from the PADB the log data about any user requests that may have affected the MSDF output results. These data must be used by the HCI module to prevent the manipulation of corrupted and/or missing data.

5.3 PADB in the Off-Line State / Performance Evaluation Mode

In the Performance Evaluation mode, the PE module has access to the overall data archived in the PADB for a given test run. The retrieved data for a given test run includes the configuration of the MSDF System used, the white and black box data produced and archived by the MSDF System and the stimulation data transferred and archived by the Data Driver component. The error reporting data generated and archived by the MSDF Test Bed and the log data about any user requests that may affect the MSDF output results must also be retrieved since these data might be needed to ensure that the Performance Evaluation module does not manipulate corrupted truth, sensor and track data. In summary, all the necessary information to assess the target tracking and identification performance of the MSDF System and its real-time performance as a real-time system must be retrieved.

In this mode of operation of the MSDF Test Bed, the modules have read-only access to the PADB.

5.3.1 Stimulation Data Retrieving

The PE module must retrieve from the PADB the stimulation data previously transferred and archived by the Data Driver component.

5.3.2 Black Box Data Retrieving

The PE module must retrieve from the PADB the MSDF track data previously transferred and archived by the MSDF Track Database component. These data are required for evaluating the input-output behavior of the MSDF system.

5.3.3 White Box Data Retrieving

If requested by the user, the PADB contains some intermediate results of interest produced by the MSDF algorithms while they were running. These data are also called "white box" data in this document. The "white box" data must be retrievable by the PE module. They describe the actual processing that has been performed by the MSDF Software (i.e., its internal behavior) to transform the input sensor and link data into the resulting MSDF tracks at the output. Appendix A provides an example of such white box data to be retrieved from the PADB.

These data are required for evaluating the internal behavior of the MSDF system and thus to have the capability of fine tuning the system.

5.3.4 Error Reporting Data Retrieving

The PE module must retrieve from the PADB the system error data previously generated and archived by the MSDF Test Bed. Once retrieved, the PE module must also report these error data to the user via the HCI module. These data should be used by the PE module to prevent the manipulation of corrupted data

5.3.5 User Requests Data Retrieving

The PE module must retrieve from the PADB the log data about any user requests that may have affected the MSDF output results. These data should be used by the PE module to prevent the manipulation of corrupted and/or missing data.

5.3.6 History of Step-by-Step Function Transitions Retrieving

The PE module must retrieve from the PADB the history of the function transitions that may have affected the MSDF real-time performance in the Step-by-Step mode. These data should be used by the PE module to prevent the evaluation of real-time performance on data generated during the manual or automatic functions of the Step-by-Step mode. These data could also be useful to explain an abrupt change in the quality of the MSDF System results if, for example, a portion of the test run has been executed using the Real-Time Like function of the Step-by-Step mode.

6.0 HUMAN COMPUTER INTERFACE (HCI) MODULE

The Human Computer Interface (HCI) module must provide all of the HCI facilities required to support all interactions of the MSDF Test Bed with its user/operator. Indeed, the HCI module must be the top-level module of the MSDF Test Bed, and as such, it must manage (i.e., integrate and control) all other modules.

Hence, through the Manager Display component, the HCI module must allow user interactions with the other modules (e.g., by prompting for values of specific configurable internal parameters for the other modules). Through the Analytical Display and PE Display components, the HCI module provides the capability to present the stimulation scenario data, the input/output data of the MSDF System, the MSDF Test Bed error/alarm data, and the results of the performance analyses of the MSDF System.

The HCI module must be designed to be intuitive, easy to use (i.e., it must be ergonomic) and modifiable. The HCI module has to use modern windowing (and other interfacing) capabilities.

6.1 HCI Module Inputs/Outputs

From the Stimulation module, the HCI module must accept the general information, ground truth and MSDF input report data to be displayed on the Analytical Display component. Similarly, the HCI module must accept the MSDF track data from the MSDF System. In the Off-Line state, all of this data must be provided to the HCI module from the PADB.

From the PE module, the HCI module must accept PE results (i.e., sensor performance and tracker performance for a given period of time) to be displayed on the PE Display component. The HCI module must also accept from a PE Result file (produced by the PE module) the MOP, RTP and tuning data to be displayed by the PE Display component.

In the Step-by-Step mode, the HCI module must accept the MSDF Busy/Ready signals from the MSDF System module. These status signals can be used by the end user to control the transfer of the stimulation file contents to the MSDF System module (and to the other modules) at the appropriate time.

Through the Manager Display component, the HCI module must provide system initialization and operation data/commands to the other modules. The HCI module must also provide the MSDF Software configuration data that allow the various MSDF algorithms and processes to configure themselves prior to any test run. Similarly, the HCI module must provide the PE configuration data that allow the various PE processes to configure themselves prior to any test run analysis.

The HCI module must archive to (and also retrieve from) the PADB the error reporting data generated by the MSDF Test Bed and the log data about any user requests that may affect the MSDF output results.

6.2 Manager Display

The Manager Display component is an interface built on top of the MSDF Test Bed software. Its main purpose is to allow the user easy access to all of the MSDF Test Bed elements or functionality.

The Manager Display component consists of a limited number of control panels (indeed, one for each of the MSDF Test Bed modes), on which the user/operator is presented with a number of options. From these control panels, the user is given different levels of access for controlling specific parameters of the various components of the MSDF Test Bed. For example, this interface must provide capabilities for:

- stimulation file selection,
- MSDF Software configuration and tuning,
- PADB management,

- PADB archive file selection,
- Analytical Display configuration,
- PE Module configuration,
- access to the MSDF Software configuration content saved in the PADB,
- PE Display configuration,
- test run launching and abortion,
- test run analysis launching and abortion,
- error reporting,
- mode transition,
- MSDF Test Bed termination,
- etc.

The control panel of any particular mode must always be accessible to the user during the mode execution. That is, no user or system action other than terminating the mode produces results that hide the control panel (or portion of it), making it inaccessible to the user.

6.2.1 User Interface Mechanism

The Manager Display must allow the user to interface with the MSDF Test Bed via Quick Action Buttons (QABs), pull-down menus, tool bars, or any other appropriate means. The QABs are each "on/off" toggle buttons that perform a specific task. If QABs are used, they must be displayed in a matrix configuration.

For any button (or section of a pull-down menu), a text label on the button must provide an intuitive description (within some appropriate size limitations) of the effect of selecting this option. Upon selection, the callback procedure property of this option is executed. Upon selection with a different button of the mouse (or the user's cursor control device), a description of the option must appear as help information for the user.

An interface mechanism (such as an Operator Guidance Readout Area) also allows the user to enter required parameter values when necessary.

6.2.2 Development Environment Mode

The HCI for the Development Environment mode must allow access/transition to the Test Bed Manager mode or to all of the development environment tools.

6.2.3 Stimulation File Selection

In the On-Line state, the Manager Display component must provide the user with a capability to select a specific MSDF scenario from the stimulation file library prior to launching a test run. No test run can be launched before this selection has been made.

6.2.4 MSDF Software Configuration and Tuning

The Manager Display component must provide the user with a capability to create/edit the MSDF Software configuration and store it in the MSDF configuration file library for future test runs.

In the On-Line state, the Manager Display component must provide the user with a capability to select a specific set of configuration parameters for the MSDF System from the MSDF configuration file library prior to launching a test run. No test run can be launched before this selection has been made. The content of the MSDF configuration file selected and used for a test run must be logged into the PADB archive file (this information could be required during the MSDF performance evaluation activities in the PE mode if the initial MSDF configuration file has, in the meantime, been modified, or if it has become unavailable).

In the PE mode of the Off-Line state, the Manager Display component must be able to extract, from the selected test run archive file, the specific set of configuration parameters used by the MSDF System to produce that test run archive file.

6.2.5 Use of Pre-Saved Setup Files

No test run can be launched before both of the selections mentioned in Subsections 6.2.2 and 6.2.3 above have been made. However, as an alternative way to perform these two selections, the Manager Display component must provide the user with a capability to create, edit and use test run setup files that pair a stimulation file with an MSDF configuration file and automatically select these files for a test run upon request by the user.

6.2.6 PADB Management

The Manager Display component must provide the user with a set of capabilities to manage the PADB activities during the execution of the MSDF Test Bed. These capabilities are described below.

6.2.6.1 Archived Data Selection

In the On-Line state, a set of default data types to be archived (containing the main data elements of interest such as the MSDF tracks, the ground truth data, etc.) must be defined and must automatically be selected by the MSDF Test Bed upon the launching of the Real-Time or Step-by-Step modes. However, the user must have the capability to select the data types archived from the list of all available data elements before the execution of a test run. As one of the possibilities, the user must have the capability to select the default set prior to conducting a test run.

6.2.6.2 Archive File Name Specification

In the On-Line state, the Manager Display component must internally specify the name of the PADB archive file to be used to archive all data produced during a given test run while in the On-Line state. The Manager Display component automatically creates a unique name for the archive file, based on the test run configuration. For example, the Manager Display component could embed in the archive file name the names of the

stimulation and MSDF configuration files used for this test run, the current time of the system, etc. Through an appropriate interface, the user must have the possibility of attaching a comment to the archive filename (e.g., "This test run shows the tracking capabilities of the MSDF System for maneuvering targets in dense clutter").

In the Off-Line state, the Manager Display component must provide the user with a capability to select a specific test run (i.e., select an archived file from the PADB library) prior to launching the test run replay and/or performance evaluation processes. No test run replay/analysis can be launched before this selection has been made.

6.2.6.3 Data Archiving Status Control

The Manager Display component must provide the user with a capability to control the status of the archiving process during the execution of the Real-Time and Step-by-Step modes. Examples of commands that must be available to the user are:

- Start Archiving
- Stop Archiving

This capability could be used to save PADB memory space when it is known to the user that some MSDF phenomenon of interest to be closely investigated happens only in a specific portion of the test run. In such a situation, it might be useless to archive the data both before and after the particular phenomenon.

6.2.6.4 Archive Statistics Displaying

In the On-Line state, the Manager Display component must display and update the archiving (i.e., the PADB) statistics required by the user to properly support the testing activities. The statistical data consist of at least the following:

i. Free disk space remaining (kilobytes).
ii. Current data reception rate (kilobytes/s).

iii. Current data archiving rate (kilobytes/s).

iv. Maximum data archiving rate (kilobytes/s).

v. Average data archiving rate (kilobytes/s).

vi. Remaining archiving time (hrs:min:s) at current archiving rate, or at local average archiving rate.

6.2.7 Analytical Display Configuration

In the On-Line state, the Manager Display component must provide the user with a capability to configure the Analytical Display component prior to the execution of a test run. This must include the disposition and size of the various display elements (e.g., position and size of the TSA), the colors to be used, etc. Figure 7 is an example of a possible Analytical Display layout.

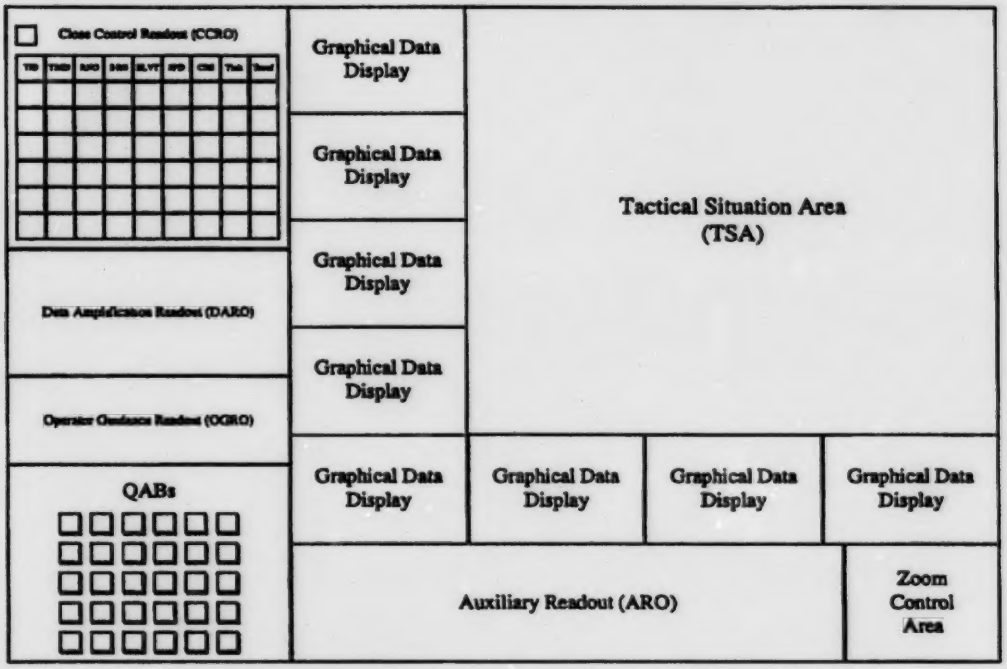

FIGURE 7 - Example of a possible Analytical Display layout

The user must also have the capability to create, edit and use display setup files that each contain some initial configuration for the display elements that corresponds to the preferences of some of the users.

6.2.8 Performance Evaluation Module Configuration

The Manager Display component must provide the user with a capability to select a specific set of configuration parameters for the PE module from the PE configuration file library prior to launching a test run analysis. No test run analysis can be launched before this selection has been made.

The Manager Display component must thus provide the user with a capability to create, edit and use PE configuration files. The PE configuration must provide the necessary options to select between:

- MSDF MOPs
- real-time MOPs
- MSDF System tuning Data

The Manager Display component must allow the user to define the volume of interest (VOI) and time interval of interest (TIOI) in the PADB for which the analysis is to be performed (Ref. 7). The TIOI specifies the portion of simulation time for which the analysis is to be performed. The VOI and TIOI may be obtained from the Replay Mode. By transitioning from the Replay Mode to the PE Mode, the user defined time slice, and the zoomed area in the shown TSA should be re-used in the PE mode as the TIOI and VOI, respectively.

The PE module configuration capability must also allow the user to select the desired MOPs and select the data (i.e.: targets, sensors and/or tracks) for which the MOPs are to be evaluated. Again, by transitioning from the Replay Mode, the hooked tracks,

targets or input reports should be used in the PE mode as the selected data on which the MOPs are to be evaluated.

6.2.9 Performance Evaluation Display Configuration

The Manager Display component must provide the user with a capability to configure the PE Display component prior to the execution of a test run analysis. This should include the disposition and size of the various display elements (e.g., position and size of a graphical plot), the colors to be used, etc.

The user must also have the capability to create, edit and use PE display setup files that each contain some initial configuration for the display elements that corresponds to the preferences of some of the users.

6.2.10 Error Reporting

Through the Manager Display component, the MSDF Test Bed must accomplish error reporting. The error reporting data must include any appropriate data that describe the anomalies that occur while the MSDF Test Bed is/was up and running. The error description data must be sufficiently detailed to allow the designer/user/operator to understand the problem and fix it if necessary. This data must include a description of the minor errors that do not cause the system to crash, but that may affect the MSDF output results.

As the bare minimum, the Manager Display component must report and display exception handling messages of the following events (these must also be logged in the PADB):

i. Receipt of invalid data from the Data Driver (e.g., a corrupted data set being sent to the MSDF System module).
ii. Receipt of invalid data from the PDBA (e.g., a corrupted data set being sent to the Performance Evaluation module).

iii. Buffer overflow: MSDF Software input FIFO / PE module.

iv. Buffer overflow: Analytical Display FIFO / PE Display.

v. PADB FIFO buffer overflow.

An area of the overall display must be used to display any error/alarm reporting message generated by the MSDF Test Bed during run time. This area must be a forced window that appears, upon the MSDF Test Bed request, in front (i.e., on top) of any other windows already in use by the program.

6.2.11 Mode Transition

The Real-Time Mode and Step-by-Step Mode control panels of the Manager Display component must provide the user with a capability to access/transition to the Test Bed Manager mode of the Control state only. Such a mode transition must be enabled only when no test run is being executed.

The Replay Mode and PE Mode control panels of the Manager Display component must also provide the user with a capability to access/transition to the Test Bed Manager mode of the Control state (such a mode transition must be enabled only when no test run analysis is being executed in the PE mode). However, since the visualization of the stimulation and MSDF data and the computation of the various measures of performance are two activities of the analysis process that are highly related, the user should have the capability to quickly switch from the Replay mode to the PE mode, and vice versa. Indeed, it is highly desirable that a direct link exist between the Analytical Display and the PE Display so that, for example, a modification of the time interval of interest on the Analytical Display automatically produces the corresponding modification on the PE Display, and vice versa. Similarly, a modification of the VOI on the PE Display should automatically produce the corresponding TSA zoom change on the Analytical Display, and vice versa.

6.2.12 Test Run Launching

In the On-Line state, the Manager Display component must provide the user with a capability to launch a test run. However, no test run can be launched before both of the selections mentioned in Subsections 6.2.2 and 6.2.3 above have been made. The user must have the possibility of selecting and using a pre-saved test run setup file that pairs a stimulation file with an MSDF configuration file and automatically selects these files for the test run (see Subsection 6.2.4 above).

In the Off-Line state, the Manager Display component must provide the user with a capability to launch a test run replay/analysis process. However, no test run replay/analysis process can be launched before the archived file selection mentioned in Subsection 6.2.6.2 above has been made.

6.2.13 Test Run Abortion

In the On-Line state, the Manager Display component must provide the user with a capability to abort an ongoing test run at any time during the test run execution. Similarly, in the Off-Line state, the Manager Display component must provide the user with a capability to abort an ongoing test run replay/analysis process at any time during the test run replay/analysis execution.

Upon such an abort request, the MSDF Test Bed must first halt all data processing, collection and archiving functions, it must then close the corresponding PADB archive and results files, and it must finally kill all of the corresponding processes (e.g., the data driver process, the MSDF process, the PE process, etc.). The MSDF Test Bed must then wait for the next user request.

6.2.14 MSDF Test Bed Abnormal Termination

A mechanism must be implemented to handle any abnormal termination event that may arise during the MSDF Test Bed program execution. When the MSDF Test Bed

detects an abnormal termination condition, it must kill all of the running processes and close all of the opened files. Upon abnormal program termination, the MSDF Test Bed must return (if possible) to the Development Environment mode of the Control state.

6.2.15 MSDF Busy/Ready Signal

The Step-by-Step Mode control panel of the Manager Display component must provide the user with a capability to monitor the MSDF Busy/Ready signals provided by the MSDF System module while the Manual function of the Step-by-Step mode is executed. Hence, an area of the Step-by-Step Mode control panel must be dedicated to the display of these status signals.

6.2.16 Stimulation Data Request

The Step-by-Step Mode control panel of the Manager Display component must provide the user with a capability to initiate a manual data request to the Data Driver component of the Stimulation module while the Manual function of the Step-by-Step mode is executed. Upon such a request, the Data Driver must accordingly send the next block of data sets. Clearly, this capability must be enabled only for the Manual function. This capability must also be enabled only when the MSDF Ready signal is received from the MSDF System module.

6.2.17 Step-by-Step Function Transitions

When the MSDF Ready signal is received and displayed, the Step-by-Step Mode control panel of the Manager Display component must provide the user with a capability to switch to the Automatic or the Real-Time Like Step-by-Step functions (instead of sending a manual data request as in section 6.2.13 above).

The Step-by-Step Mode control panel of the Manager Display component must also provide the user with a capability to specify, before switching to a different function,

a simulation time period during which the Data Driver will remain in that new specified function.

While a portion or all of the test run is being performed in the new specified function, the user must have the possibility, through a synchronous user input through the Step-by-Step Mode control panel of the Manager Display component, of switching the new specified function back to the Manual Step-by-Step function even if the requested simulation time period has not been completed.

A synchronous user input means that the processing of the current data set by the MSDF system has to be completed before the user request is acknowledged.

6.3 Analytical Display

The Analytical Display must be designed to support the evaluation of the MSDF algorithms by showing images of the true target characteristics, sensor contacts, clutters and false alarms, MSDF track characteristics, etc., generated by the other modules. It must provide for a global view of the situation and allow a quick assessment of the MSDF System performance. The main purpose of the Analytical Display component is thus to "show" the data of interest to the end user, not to "control" the various processes of the MSDF Test Bed. The Manager Display component discussed above must address all of the system control functionality.

The Analytical Display must receive its data and display them according to the current selections of the user. It must include the appropriate graphics and alphanumeric elements required to present the data along with the interface mechanism that allows the user to manipulate these elements. Figure 7 is an example of a possible Analytical Display layout. The display management interface must be interactive and user friendly. All of the required operations must be performed using push buttons, toggle buttons, popup menus, scroll bars, arrow buttons, etc., as appropriate.

A number of data presentation capabilities have been grouped under the "Analytical Display" label. Examples of these capabilities are the Analytical Display management, Tactical Situation Analysis (TSA) displaying, input element hooking, displaying of various data (e.g., graphical, close control, amplification and auxiliary data), errors/alarms reporting, etc. These capabilities are discussed next.

6.3.1 Data Reception

In the On-Line state, the Analytical Display component receives all data from the Data Driver component of the Stimulation module and the MSDF Track Database component of the MSDF System module. This data consist of the following elements:

Stimulation Data

- General Information
- MSDF Input Reports
  - Contact Data
  - Track Data
  - Navigation Data
- Ground Truth Data

Black Box Data

- MSDF Tracks

In the Replay mode of the Off-Line state, the Analytical Display component receives the same data from the PADB (i.e., archived data).

6.3.2 Windowing Interface

The MSDF Test Bed must allow windows to be opened, with the capability of resizing and repositioning all windows, in order to satisfy the Analytical Display component requirements. This HCI windowing interface must contain the Tactical

Situation Analysis (TSA) display, all Graphical Data displays, etc., when these elements are activated (i.e., upon user request).

One has to define the maximum number of windows allowed since this number has an impact on the real-time performance of the system.

6.3.3 Analytical Display Management (User Requests/Actions)

A mechanism, such as Quick Action Buttons (QABs), pull-down menus, tool bars, or any other appropriate means, must allow the user to manage the Analytical Display component. The QABs are each "on/off" toggle buttons that perform a specific task. If QABs are used, they must be displayed in a matrix configuration.

For any button (or section of a pull-down menu), a text label on the button must provide an intuitive description (within some appropriate size limitations) of the effect of selecting this option. Upon selection, the callback procedure property of this option is executed. Upon selection with a different button of the mouse (or the user's cursor control device), a description of the option must appear as help information for the user.

6.3.4 Tactical Situation Analysis (TSA) Displaying

One of the most important elements of the Analytical Display is the TSA. It contains a graphical representation of the situation surrounding ownship. All targets being tracked by the MSDF System (or any other tactical and scenario related information) must be represented graphically in the TSA according to the attributes chosen by the user. The TSA must also have a background map. The user must have the capability to toggle on and off the various symbols and graphics, select relevant information, zoom into specific areas, rotate the scene, etc. Both a three-dimensional perspective view and a two-dimensional radar-like view must be provided. The radar-like view must also support both Cartesian and polar axes.

6.3.4.1 MSDF Track Presentation

Upon a user request, the MSDF tracks must be displayed on the TSA with graphical attributes selected by the user. The Analytical Display component must allow the user to toggle on or off, via the Analytical Display Management capability, all MSDF tracks simultaneously. The standard NTDS symbology in 2D (STANAG 4420) should be used by the Analytical Display component to present the MSDF tracks.

6.3.4.1.1 All MSDF Tracks

Each MSDF track should possess the following graphical attributes:

i. Track Symbol
ii. Velocity Leader
iii. Area Of Uncertainty (AOU) / Gates
iv. Track Tag
    - Track Number
    - Target Identification (Highest Belief)
    - Track Quality
    - Range
    - Bearing
    - Altitude
    - Course
    - Speed
    - Time
v. Track History (On-Line state only)

The user must have the possibility of turning each MSDF track attribute on or off (for all the MSDF tracks simultaneously) through the Analytical Display Management capability. These changes must override currently selected graphical attributes for the hooked MSDF tracks.

6.3.4.1.2 Hooked MSDF Tracks

All hooked[1] MSDF tracks must be graphically displayed with graphical attributes selected by the user. For each of the following MSDF track attributes, the user must have the possibility of locally overriding the global MSDF track attributes set for all MSDF tracks through the Analytical Display Management capability:

i. Track Symbol
ii. Velocity Leader
iii. Area Of Uncertainty (AOU) / Gates
iv. Track Tag
    - Track Number
    - Target Identification (Highest Belief)
    - Track Quality
    - Range
    - Bearing
    - Altitude
    - Course
    - Speed
    - Time
v. Track History (On-Line state only)

Upon user request, the HCI module must display the identification probability tree of any hooked MSDF track in the Amplification Data display (see Subsection 6.3.8). In the Replay mode of the Off-Line state, the user must also have the additional capability of specifying a simulation time instant (contained in the simulation time interval currently presented on the TSA) for the display of the identification probability tree.

[1] The user can "hook" specific objects (e.g., a particular track, a given input report) on the TSA by individually selecting them through some user interface mechanism, in order to highlight their distinctive characteristics.

6.3.4.2 Input Report Presentation

Upon user request, the input report data, i.e.:

i. true target contacts,
ii. false contacts, and
iii. link tracks,

must be displayed on the TSA with graphical attributes selected by the user. The Analytical Display component must allow the user to toggle on or off, via the Analytical Display Management capability, all input report data of a given type (e.g., all false contacts) simultaneously. One has to define an appropriate symbology in 2D to be used by the Analytical Display component to present the input report data.

6.3.4.2.1 All Input Reports

Each input report element should possess the following graphical attributes:

i. Input Report Element Symbol
ii. Velocity Leader (if appropriate for the element type)
iii. Area Of Uncertainty (AOU) / Gates (if appropriate for the element type)
iv. Input Report Tag
    - Source
    - Range
    - Bearing
    - Altitude
    - Course (if appropriate for the element type)
    - Speed (if appropriate for the element type)
    - Time
v. Track History (On-Line state only, if appropriate for the element type)

The user must have the possibility of turning each input report element attribute on or off (for all the input report elements of a given type simultaneously) through the Analytical Display Management capability. These changes must override currently selected graphical attributes for the hooked input report elements.

6.3.4.2.2 Hooked Input Reports

All hooked input reports must be graphically displayed with graphical attributes selected by the user. For each of the following input report element attributes, the user must have the possibility of locally overriding the global input report element attributes set for all input reports of a given type through the Analytical Display Management capability:

i. Input Report Element Symbol
ii. Velocity Leader (if appropriate for the element type)
iii. Area Of Uncertainty (AOU) / Gates (if appropriate for the element type)
iv. Input Report Tag
    - Source
    - Range
    - Bearing
    - Altitude
    - Course (if appropriate for the element type)
    - Speed (if appropriate for the element type)
    - Time
v. Track History (On-Line state only, if appropriate for the element type)

6.3.4.3 Ground Truth Presentation

Upon a user request, the ground truth targets must be displayed on the TSA with graphical attributes selected by the user. The Analytical Display component must allow the user to toggle on or off, through the Analytical Display Management capability, all

ground truth targets simultaneously. One has to define an appropriate symbology in 2D to be used by the Analytical Display component to present the ground truth targets.

6.3.4.3.1 All Ground Truth Data

Each ground truth target should possess the following graphical attributes:

i. Ground Truth Target Symbol
ii. Velocity Leader
iii. Ground Truth Target Tag
    - Target Number
    - True Target ID
    - Range
    - Bearing
    - Altitude
    - Course
    - Speed
    - Time

The user must have the possibility of turning each ground truth target attribute on or off (for all the ground truth targets simultaneously) through the Analytical Display Management capability. These changes must override currently selected graphical attributes for the hooked ground truth targets.

6.3.4.3.2 Hooked Ground Truth Data

All hooked ground truth targets must be graphically displayed with graphical attributes selected by the user. For each of the following ground truth target attributes, the user must have the possibility of locally overriding the global ground truth target attributes set for all ground truth targets through the Analytical Display Management capability:

i. Ground Truth Target Symbol

ii. Velocity Leader

iii. Ground Truth Target Tag

- Target Number
- True Target ID
- Range
- Bearing
- Altitude
- Course
- Speed
- Time

6.3.4.4 Scaling and Centering (Zooming)

Through the windowing interface, the Analytical Display component must allow the zooming in or out of any area in the TSA through a user defined scale and center. Scaling must be through a menu interface with discrete values and centering must be defined with respect to one of the following:

i. Ownship

ii. Track Number

iii. Fixed Position

a. Latitude Longitude

b. Cursor

iv. Anchored Cursor

An option should allow the user to define directly on the TSA an equally proportioned magnification area through the mouse interface. The HCI module must auto-center and auto-scale the image. The magnification area must be defined either:

i. corner to corner, or

ii. center to corner.

6.3.4.5 Coordinate System Selections

The Analytical Display component must allow the user to select, through the Analytical Display Management capability, the coordinate system to be used for data presentation on the TSA.

6.3.4.5.1 2D/3D Selection

The Analytical Display component must allow the user to toggle between the 2D and 3D scales on the TSA through the Analytical Display Management capability.

6.3.4.5.2 2D Option- Change Axes

When 2D scales are selected for the TSA, the Analytical Display component must allow the user to toggle between the Cartesian and Polar axes through the Analytical Display Management capability.

6.3.4.5.3 3D Option - Rotate 3D Perspective

When displaying the data in 3D on the TSA, the Analytical Display component must allow the user to select an observation point in space to define the perspective of the display. The Analytical Display component should be capable of rotating the 3D display around the X, Y, and Z axes via the windowing interface. The Analytical Display component should also be able to return to the initial observation point from any point.

6.3.4.6 Geographic Information System (GIS)

A geographic information system should be used to display background maps on the TSA.

6.3.5 Input Element Hooking

The Analytical Display component must allow the user to hook a number of input elements for the purpose of highlighting the information they contain. Based on system performance and practical display layout considerations, one has to determine the maximum number of input elements that can be hooked simultaneously by the user.

6.3.5.1 MSDF Track Hooking

The user should have the capability to hook MSDF tracks directly on the TSA (with the mouse or the user's cursor control device), or through the Analytical Display Management capability.

6.3.5.2 Input Report Hooking

The user must have the capability to hook input reports (either input contacts or input tracks) through the Analytical Display Management capability.

6.3.5.3 Ground Truth Data Hooking

The user must have the capability to hook ground truth targets through the Analytical Display Management capability.

6.3.6 Graphical Data Displaying

The user must have the capability to display, in real-time, graphical data (e.g., MSDF track range versus time, true target speed versus true target range, etc.) in specialized graphical data windows of the Analytical Display component. The information to be potentially displayed includes the stimulation data and the black box data. The graphical data corresponding to hooked elements only (i.e., MSDF tracks, input reports, ground truth targets) should be displayed.

The user must have the capability to select the particular combination of data types to be displayed in a given Graphical Data Display window (e.g., range vs. time, altitude vs. range, speed vs. bearing, etc.) from all of the available data types, at any time during the MSDF Test Bed program execution.

The HCI module must allow the user to scale, at any time during program execution, both axes of the graphical data display.

6.3.7 Close Control Data Displaying

The Close Control Readout (CCRO) area must provide detailed information about hooked elements. Once the user has hooked an element (e.g., an MSDF track), the CCRO area must display detailed information about that element. For an MSDF track, such information consists of the track number, values for the target range, bearing, course, speed, etc.

In the Replay mode of the Off-Line state, the user must have the additional capability to specify a specific simulation time instant (contained in the simulation time interval currently presented on the TSA) for the display of the CCRO data.

6.3.8 Amplification Data Displaying

The Data Amplification Readout (DARO) area must display status or more detailed data about hooked elements and functions performed by the system.

For example, once the user has hooked an MSDF track, the DARO area must display detailed information about the identity of the target. Assuming that a Dempster-Shaffer approach to target identification is used (Ref. 8), a portion of this area must display the belief held by the MSDF System for specific target attributes. For example, a column could contain information concerning friend, foe and neutral. The system must display these values in descending order of belief so that the attribute with the highest belief always appears on the first row and can more easily be seen. Another portion of the

DARO area should consist of a scrollable window which should present the MSDF System's suggestions for the identification of the hooked MSDF track. The first line of the scrolled window must display the ignorance or uncertainty left in the MSDF system. The second (and any following) line must contain the belief followed by a proposition which is a list of actual target identifications. The user must have the capability to use the scroll bar to see the complete text.

In the Replay mode of the Off-Line state, the user must have the additional capability to specify a specific simulation time instant (contained in the simulation time interval currently presented on the TSA) for the display of the amplification data.

6.3.9 Auxiliary Data Displaying

The Auxiliary Readout (ARO) area should primarily contain navigation and ownship status information. For example, the ship's latitude, longitude, course and speed could be displayed in this area. The simulation time (in the On-Line state only) must also be displayed in the ARO.

In the Replay mode of the Off-Line state, the user must have the additional capability to specify a specific simulation time instant (contained in the simulation time interval currently presented on the TSA) for the display of the auxiliary data.

6.3.10 Errors/Alarms Reporting

An area of the overall display must be used to display any system message (such as error or alarm messages) generated by the MSDF Test Bed during run time. This area must be a forced window that must appear, upon the MSDF Test Bed request, in front (i.e., on top) of any other windows already in use by the program.

6.3.11 Time Slice Mechanism

In the Replay mode of the Off-Line state, the Analytical Display component must allow the user to specify a time slice (i.e., a simulation time window, or a time filter, with a user definable width ΔT) that should define the scope of the archived test run data to be retrieved from the PADB and displayed simultaneously on the Analytical Display. Hence, any input information data contained in the archive file of the PADB that fall within the time slice must be presented on the Analytical Display according to the current graphical attribute selections of the user. The user must have the possibility to position this time slice anywhere on the test run time line.

6.3.11.1 Time Slice Width

The Analytical Display component must allow the user to define the time slice width ΔT via the Analytical Display Management capability. Through the selection of an appropriate value for ΔT, the user should have the possibility of presenting all of the test run data simultaneously on the Analytical Display, according to the current graphical attribute selections.

6.3.11.2 Time Slice Repositioning

The Analytical Display component must allow the user to define a time tag (T) that should be anchored to the upper most limit of the time slice ΔT (i.e., the time slice should extend from T-ΔT to T). This time tag can be used to reposition the time slice anywhere on the test run time line using the following capabilities:

i. Play (move the time slice (i.e., move the time tag T) backward/forward at a user definable rate).
ii. Pause.
iii. Rewind (play backward at twice the current play rate, up to a maximum of 4 times).

iv. Forward (play forward at twice the current play rate, up to a maximum of 4 times).

v. Jump (direct jump to a different, valid time tag definable by the user).

6.4 Performance Evaluation Display

The PE module described in Chap. 7.0 is the necessary complement to the above mentioned Analytical Display component. It includes a set of tools to help the user in his/her quantitative assessment of the performance of the MSDF algorithms and techniques. These tools comprise a compilation mechanism for the tracking statistics, computation mechanisms for the various measures of performance, etc.

With respect to the HCI module, the PE results must be presented in a user-friendly manner to the end user through the PE Display component. This includes such things as the output of MSDF performance results in summary tables and listings, the plotting of the tracking statistics, etc.

The PE display must have the necessary capabilities to concurrently display any combination of MOPs specified by the user with respect to the defined VOI and TIOI. The results of the PE module should be displayed in the appropriate format for each particular MOP, either as tables or as graphics. These outputs must be exportable (e.g., as encapsulated postscript files) and printable.

The PE Display component must receive its data from the PE results files. The PE display must restore from the PE results files the analysis results with the complete setup that has been used to obtain these results. The complete setup can be defined as the PE configuration content, the archive file name of the test run for which the MOP has been evaluated, and the MSDF System configuration content that generated the test run archive.

6.4.1 Table Data Displaying

The user must have the capability to display text data from the PE results files in specialized text table data windows of the PE Display component. The information to be displayed includes the stimulation data, the black box data, the white box data, the MSDF MOPs and the real time MOPs.

The user must have the capability to select the particular combination of MOPs to be displayed in a given text table data display window, for the selected VOI and TIOI. The HCI module must allow the user to export the text data display to a file or to a printer. The PE Display component must provide the user with a capability to present RTP and tuning data on text table displays.

6.4.2 Graphical Data Displaying

The user must have the capability to display graphical data from the PE results files in specialized graphical data windows of the PE Display component. The information to be displayed includes the stimulation data, the black box data, the white box data, the MSDF MOPs and the real-time MOPs.

The user must have the capability to select the particular combination of MOPs to be displayed in a given Graphical Data Display window, for the selected VOI and TIOI. The HCI module must allow the user to scale both axes of the graphical data display and to export the graphical data display to a file or to a printer. The PE Display component must provide the user with a capability to present RTP and tuning data on graphical displays.

7.0 PERFORMANCE EVALUATION MODULE

The main purpose of the Performance Evaluation (PE) module is to provide a capability that allows the end user to assess if the MSDF System is working properly, as expected. There are two major aspects that must be taken into account to perform this assessment:

1) the performance of the MSDF System as a target tracking and identification process (i.e., does the MSDF System compute the "correct", "expected" output given the stimulation data provided?), and
2) the performance of the MSDF System as a real-time system (i.e., is the MSDF output provided in a timely manner with respect to the real-time constraints specified for the system?).

The first PE aspect mentioned above is really about the essence of the MSDF System. One has to assess the performance of the MSDF System at tracking and identifying targets, independently of the time actually taken by the MSDF System to perform these two activities.

The objective of the overall system (i.e., the system comprising both the sensors and the MSDF elements) is to support the perception of the external situation as efficiently and accurately as possible. During its operation, the overall system generates both a measured tactical picture (i.e., the output of the sensing process, here provided as part of the stimulation data) and an estimated picture (embodied in the tracks stored in the MSDF Track Database and whose quality should in principle be superior to that of the measured picture). The purpose of the MSDF performance evaluation process is to evaluate the ability of the MSDF System to generate an estimated tactical pictures that accurately reproduce the ground truth tactical picture (Refs. 7 and 9). The evaluation objective is to quantify the potential divergence that may result from the many limitation factors affecting the performance of the MSDF System.

To support the MSDF performance assessment activity, the PE module must allow the user to compute various MSDF Measures Of Performance (MOPs). These MOPs measure various aspects of the system behavior. The MOP problem for an MSDF system can be stated as follows. Given the output of an MSDF algorithm, how does one determine the "performance" of the algorithm? The answer depends on one's purpose and leads to distinctions between types of MOPs. The types of MOPs typically considered (Ref. 7) are those which measure how well an MSDF algorithm is performing against tactically crucial parameters (e.g., localization radial miss distance). These are typically of great interest to the user community.

Practicality dictates that several different MOPs should be defined to cover several aspects of MSDF performance at a given time. Typically, a finite set of MOPs is considered in order to illuminate the following three major aspects of target tracking and identification performance (Ref. 7):

- reaction time,
- track quality,
- target identification accuracy.

Reaction time is defined as the time taken by the MSDF System to output a useful tracking or identification result about objects entering in the tactical volume of interest. The measures of individual track quality focus on three aspects of the tracking process: (1) track estimation, (2) track purity (i.e., correlation quality), and (3) track management. And finally, the purpose of classification MOPs is to score how well the MSDF algorithm is identifying targets.

The Performance Evaluation module must also allow the end user to establish the real-time requirements of the MSDF System on speed, timeliness, responsiveness, predictability, graceful adaptation, etc. As is well known, the correctness of a real-time system depends not only on the logical and accurate results of the computations (as

assessed using the MSDF MOPs discussed above), but also on the time at which the results are produced. One has to identify and define an appropriate set of measures of Real-Time Performance (RTP) of the MSDF System. One also has to determine all data collection activities that would need to be conducted during the operation of the MSDF System so as to obtain estimates of the RTP measures.

Finally, especially during the MSDF Software development activities, multiple test runs will be performed in order to tune the various MSDF algorithms to both achieve the desired target tracking and identification performance and meet the real-time constraints for the system. Hence, the PE module must provide a capability to support the tuning of the MSDF algorithms. This is mostly concerned with the appropriate formatting and displaying of the white box data generated by the MSDF Software upon user request.

In summary, the PE module must comprise a set of tools to support the designer/user/operator in his/her quantitative assessment of the performance of the MSDF algorithms and techniques. These tools include a compilation mechanism for the tracking statistics, computation mechanisms for the various measures of performance, etc.

The PE module must compute the requested measures of performances and store the analysis results, with the complete setup that has been used to obtain these results, in the PE results files. The complete setup can be defined as the PE configuration contents, the archive file name of the test run for which the MOPs have been evaluated, and the MSDF System configuration contents that have been used for the generation of the test run archive.

7.1 PE Module Inputs/Outputs

To achieve the performance assessment activities described above for a given test run, the PE module must have access to the overall data archived in the PADB for this test run. In particular, the PE module requires the general information, ground truth and MSDF input reports data archived in the PADB by the Stimulation module.

The general information data set provides useful parameters such as the duration of the test run. The ground truth tactical picture represents the real composition and status of a scenario of tactical interest. It serves as a basis for comparison with the measured and estimated tactical pictures. It depicts the known activities (i.e., position, kinematic behavior, emissions, and identity) of a known number of real, distinct objects (targets) in a given area of interest. It is assumed that the MSDF evaluator knows at any time all the characteristics and behavior of the set of targets which are actually part of ground truth. The MSDF input reports (i.e., the measured tactical picture) are the known inputs that were submitted to the MSDF Software for processing.

The PE module also requires both the white and black box data generated during the test run and archived in the PADB. The white box data is used mostly as MSDF algorithm tuning information. Clearly, all error reporting and user request data that may affect the output of the MSDF System are important for the PE module since they have a direct impact on the end user conclusions about the system performance. These data have thus to be retrieved from the PADB.

The PE module must accept system initialization (e.g., the PADB archive file name, the PE configuration file name, etc.) and operation (e.g., test run analysis launching, abortion, etc.) data/commands from the HCI module. The PE module must also accept the PE configuration data that allow the various PE processes to configure themselves prior to any test run analysis. These configuration data include such things as the user settings for the tactical volume of interest (VOI) and the time interval of interest (TIOI), the requested MOP, RTP or tuning data, and the targets, sensors or tracks on which the measures of performance are to be evaluated. The PE module must provide, via the PE Result files, the resulting MOP, RTP and tuning data to the PE Display component of the HCI module for presentation to the end user.

7.2 MSDF Measures Of Performance (MOPs)

The Performance Evaluation module must have the required capabilities to compute a finite set of MSDF measures of performance (MOPs). The Performance Evaluation module must then store the results in the Performance Evaluation results files. As mentioned earlier, all Performance Evaluation result files archived must contain the complete setup that has been used to obtain these results. Typical MSDF MOPs that can be selected for the MSDF testbed illuminate the following three major aspects of the tracking and identification performance of the MSDF System (Ref. 7):

- reaction time,
- track quality, and,
- identification accuracy.

For all or selected sensors, all or selected targets and all or selected tracks in the VOI, the PE module must have the capability to calculate these MOPs (or similar MOPs) upon user request and based on the data retrieved from the PADB.

The tracks are either directly selected (by specifying the appropriate track numbers) or indirectly selected. Since the computation of some MOPs involves both track and ground truth elements, different ways to link these two must be offered to the user of these MOPs. A track or a target must be selected from the available tracks and targets in the VOI for the time interval of interest:

- If a specific track is first selected, the user must then specify the target(s) in one of the following ways to produce the required analysis.

    i) The MOP of the track is computed for each track update instant with the ground truth that is at the origin of that track update. The result is a single graphic showing the MOP of the track with any ground truth that was the origin of a track update at a given time.

ii) The list of all targets that have generated contacts used to update the track is presented to the user. From the list, a target must be selected by the user. The MOP of the track is computed for each track update instant with the ground truth associated to the selected target. The result is a single graphic showing the MOP of the track with the ground truth associated to the selected target.

iii) The list of all targets that have generated contacts used to update the track is made. The MOP of the track is computed with each target of the list each time the track is updated. The result is a graphic with several plots. Each plot represents the track's MOP computed for a given target.

- If a specific target is first selected, the user must then specify the track(s) in one of the following ways to produce the required analysis.

i) Every time this target has generated a contact used to update a track, the MOP is computed. The result is a single graphic showing the MOP of any track computed with the selected target.

ii) The list of all tracks that have used contacts from this target to get updated is presented to the user. From this list, the user must specify a track. The MOP of that track is computed for each track update instant by comparing it with the ground truth associated to the selected target.

iii) The list of all tracks that have used contacts from this target to get updated is made. The MOP of each track of the list is computed with the selected target. The result is a graphic with several plots. Each plot represents a given track's MOP computed for the selected target.

For some other MOPs, the user must specify directly the tracks for which these MOPs are to be evaluated.

7.3 Real-Time System Performance Measures

The Performance Evaluation module must have the capability to compute a finite set of real-time performance measures (RTPs), and store the results in the PE results files. As mentioned earlier, all saved PE result files must contain the complete setup that has been used to obtain these results.

The real-time system performance measures must allow the end user to establish the real-time requirements of the MSDF System on speed, timeliness, responsiveness, predictability, graceful adaptation, etc. As is well known, the correctness of a real-time system depends not only on the logical and accurate results of the computations (as assessed using the MSDF MOPs discussed above), but also on the time at which the results are produced. One has to identify and define an appropriate set of measures of Real-Time Performance (RTP) for the MSDF System. One has also to determine all data collection activities that would need to be conducted during the operation of the MSDF System so as to obtain estimates of the RTP measures.

7.4 MSDF System Tuning

The PE module must have the capability to extract selected information from the archived "white box" data, and store the results in the PE results files. As mentioned earlier, all saved PE result files must contain the complete setup that has been used to obtain these results.

Especially during the MSDF Software development activity, multiple test runs will be performed in order to tune the various MSDF algorithms in order to both achieve the desired target tracking and identification performance and meet the real-time constraints for the system. Hence, the PE module must provide a capability to support the tuning of the MSDF algorithms. This is mostly concerned with the appropriate formatting and displaying of the white box data generated by the MSDF Software upon user request.

The MSDF System tuning process is highly design dependent and its detailed definition is thus out of the scope of this document.

8.0 CONCLUSION

Multi-Source Data Fusion (MSDF) is about tactical picture compilation. It is concerned solely with individual entities (aircraft, missiles, etc.) considered in isolation (i.e., groups or formations of entities are not considered at this fusion level). An MSDF system processes the data and information reported by multiple dissimilar sources in order to correctly and quickly derive the best estimates of the current and future kinematic properties for each hypothesized (or perceived) entity in the operational environment, and to develop inferences as to the identity and key attributes of these entities.

Research and development activities were initiated several years ago at DREV to explore sensor management, integration and MSDF concepts that could apply to the current AWW sensor suite of the HALIFAX class ships, as well as their possible future upgrades, in order to improve their performance against anticipated future threats.

The approach retained for these R&D activities has been to employ sufficiently representative sensor and phenomenological simulations and, consequently, a highly modular, structured, and flexible simulation environment, the CASE_ATTI test bed, has been developed as a proof-of-concept demonstrator for MSDF. In addition to high-fidelity stimulation, one of the main requirements of the CASE_ATTI testbed has been to provide the algorithm-level test and replacement capability required to study and compare the applicability and performance of advanced, state-of-the-art MSDF techniques.

However, the MSDF systems foreseen to support command teams on board Canadian warships must be real-time systems, and the correctness of the final implementation of a real-time system is not only a function of the accuracy of the computations performed by the system, but also a function of the time at which the results of the computations are produced. Unfortunately, even though CASE_ATTI is a very powerful tool, it does not provide all of the capabilities required to meet all of the performance assessment needs for MSDF systems. CASE_ATTI is a non-real-time proof-

of-concept simulation environment. Therefore, a separate means is required to investigate the real-time aspects of the MSDF systems proposed for the future.

This document gave a detailed description of such a means that will help to establish the real-time requirements of MSDF systems on speed, responsiveness, predictability, timeliness, etc. At the same time, the MSDF Test Bed will also provide a capability to evaluate various measures of performance for these MSDF Systems.

An overview description of the MSDF Test Bed in terms of the main interacting elements (i.e., the modules) that compose the system was first given. Then, the purpose of each module was described along with how these modules work and interact together to achieve the overall functionality and objectives of the MSDF Test Bed. However, not all of these modules are used simultaneously during the execution of the MSDF Test Bed. Different operating conditions involve different modules. Hence, an overview of the MSDF Test Bed in terms of system states and operating modes that complements the decomposition into modules and components was given. Finally, detailed descriptions of the Stimulation, Performance Analysis Database (PADB), Human Computer Interface (HCI) and Performance Evaluation (PE) modules were provided.

The MSDF test bed proposed in this document will be a major component of a set of tools relevant to the development and/or acquisition of integrated shipboard C2 systems for the Canadian Patrol Frigate (CPF). It is a versatile test bed that can serve as a tool for extensive exploratory and empirical analyses and evaluation of theoretical concepts that are fundamentally important to achieving MSDF. It will provide a serious basis for evaluating our past R&D progress and for making important tradeoff decisions related to our future efforts in MSDF. For example, we shall be in a position to provide well-founded answers to fundamental questions like: what are the individual and collective performances of these MSDF processes that can be realistically automated in real-time?; how do we know that our work on theoretical MSDF concepts is leading to realizable performance improvements on board the ship and what is their impact (both

absolute and relative) on improving mission success?; in short, how do we know we are making progress, and at what cost? While this is clearly an ambitious path to follow, it is an essential and much needed one.

9.0 ACKNOWLEDGEMENTS

The author would like to acknowledge the contribution of Dr. Éloi Bossé, LCdr Jerry McLean, Messrs. Jean-Pierre Lachance, Nicolas Duclos-Hindié and Toros Aledjian to the definition of the MSDF test bed presented in this document.

10.0 REFERENCES

1. Roy, J.M.J. and Bossé, É., "Sensor Integration, Management and Data Fusion Concepts in a Naval Command and Control Perspective", DREV-R-9816, October 1998, UNCLASSIFIED

2. Roy, J.M.J. and Bossé, É., "A Generic Multi-Source Data Fusion System", DREV-R-9719, June 1998, UNCLASSIFIED

3. Roy, J.M.J., Bossé, É. and Dion, D., "CASE_ATTI: An Algorithm-Level Testbed for Multi-Sensor Data Fusion", DREV R-9411, May 1995, UNCLASSIFIED

4. Bossé, É., Duclos-Hindié, N., Roy, J.M.J. and Dion, D., "Realistic Simulations of Surveillance Sensors in an Algorithm-Level Sensor Fusion Testbed", SPIE Proceedings, Vol. 2561, Signal and Data Processing of Small Targets 1995, San Diego, 11-13 July 1995, UNCLASSIFIED

5. Bossé, É., Dion, D. and Roy, J.M.J., "A Shipborne Surveillance Radar Model for Data Fusion Studies", DREV R-9429, November 1995, UNCLASSIFIED

6. Roy, J.M.J., Duclos-Hindié, N., Triki, Z., Dessureault, D. and Bossé, É., "Generation of Realistic Stimulation Data for the Demonstration of a Real-Time Multi-Source Data Fusion System for the Halifax Class Ships (Version 3.0)", DREV TR 1999-213, September 1999, UNCLASSIFIED

7. Bossé, É. and Roy, J.M.J., "Measures of Performance for the ASCACT Multi-Sensor Data Fusion Demonstration Program", DREV-TM-9618, March 1997, UNCLASSIFIED

8. Bossé, É. and Roy, J.M.J., "Fusion of Identity Declarations from Dissimilar Sources Using the Dempster-Shafer Theory", Optical Engineering, Volume 36, Number 3, March 1997.

9. Roy, J.M.J. and Bossé, É., "Definition of a Performance Evaluation Methodology for Sensor Data Fusion Systems", DREV R-9423, September 1995, UNCLASSIFIED

APPENDIX A

EXAMPLES OF BLACK AND WHITE BOX DATA

A.1 Introduction

This appendix provides definitions and examples of the white and black box data produced by the MSDF System module of the MSDF Test Bed.

A.2 Black Box Data Definition

The black box data are the standard output results produced by the algorithms of the MSDF Software and normally provided to the final user/operator of the MSDF System. The expression "Black Box Data" is used here to denote that the MSDF Software is considered for performance evaluation from the viewpoint of its external (i.e., I/O) behavior only. Some known inputs are provided to the MSDF Software that produces some (hopefully expected) outputs. These outputs are tagged as "Black Box Data" since no information is available about the actual processing that has been performed by the MSDF Software to transform the known inputs into the observed outputs. Hence, the MSDF Software is perceived as a black box.

During its operation, the MSDF Software component generates an estimated tactical picture that should accurately reproduce the ground truth tactical picture. This perception of truth by the MSDF Software is embodied in the tracks that are established and maintained as the sensors sample the environment. Ideally, the MSDF Software should acquire and maintain a number of unambiguous, stable tracks that corresponds exactly to the number of objects of interest in the physical environment.

Typically, each track record in the MSDF Track Database includes a state vector (with a corresponding covariance matrix) estimating the target kinematic properties (i.e., position, velocity, acceleration, etc.) in a coordinate system common to all sensors, and one or more propositions about non-kinematic properties (target type, class, identity,

radar cross-section, etc.) of the target, each with its associated likelihood function. The information kept in a track record also includes a time tag corresponding to the last update time, the quality of the track, the blip/scan parameter for each sensor, etc.

The detailed definition of this type of data is design dependent. Nevertheless, an example of black box data is provided next.

A.2.1 Black Box Data Example (MSDF Track Data Record)

Bookkeeping / Administrative Parameters

Sector Number (Integer)
Track Identification Number (Integer)

Track Management Parameters

Track Status (Integer)
Track Quality / Score (Float)

Time Parameters

Time of Last Measurement Update (s: Float)

Contact Update History

Blip/Scan [s] (one byte: 0/1 for each source)

Estimated Kinematics Parameters (Last Update)

Estimated Position in "x" (m: Float)
Estimated Position in "y" (m: Float)

Estimated Velocity in "x" (m/s: Float)
Estimated Velocity in "y" (m/s: Float)

State Estimation Covariance P (4x4 Matrix: Float)

Altitude / Height (m: Float)

Estimated Identification Parameters (Generic ID)

Type [GID]	(beliefs of the various types (e.g., unknown, missile, fighter, etc.)
Allegiance [GID]	(beliefs of the various allegiances (e.g., unknown, friend, foe, neutral)
Country [GID]	(beliefs of the various countries (e.g., unknown, Canada, USA, etc.)
Lethality [GID]	(beliefs of the various lethalities (e.g., unknown, low, medium, etc.)
Threat [GID]	(beliefs of the various threats (e.g., unknown, low, medium, etc.)

Estimated Identification Parameters (Specific ID)

Propositions [p]	About Specific Platforms (Including the Ignorance)
Mass [p]	Associated With Each Specific Proposition

A.3 White Box Data Definition

The white box data are some internal intermediate results of interest produced by the algorithms of the MSDF Software while it is running (see Fig. 8). The white box data describe the actual processing that has been performed by the MSDF Software to transform the known inputs into known outputs. The expression "White Box Data" is thus used here to denote that the MSDF Software is considered for performance evaluation from the viewpoint of its internal behavior (i.e., procedure execution). Hence, the MSDF Software is perceived as an open or white box.

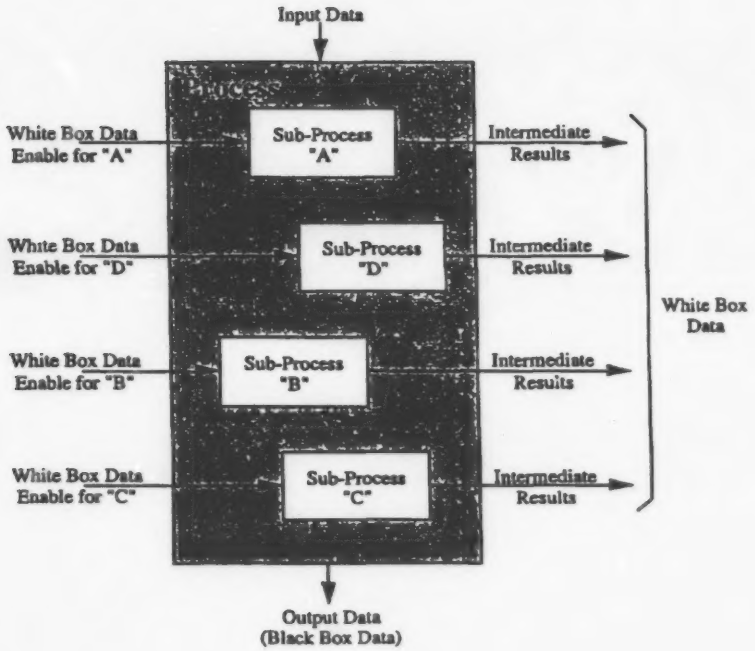

FIGURE 8 - Illustration of the white box data principle

This data is mostly required to perform detailed performance evaluation analyses during the development and tuning of the MSDF Software. Some internal results can be useful if one needs more information about a specific parameter used to compute a final output result that is questionable. When the MSDF Software is perfectly tuned and operational, the white box data should not normally be provided to the final user/operator of the MSDF System.

Clearly, the real-time performance of the MSDF Software can be affected when some parameter values are probed inside the MSDF System while it is running. Hence, this data must be extracted only when strictly necessary, and the MSDF output results must then be analyzed with great care.

White box data is highly design dependent and its detailed definition is thus out of the scope of this document. Nevertheless, an example of white box data is provided next.

A.3.1 White Box Data Example

The example is a printed output of typical white box data that could be extracted from a Multiple Hypothesis Tracking (MHT) algorithm while it is running to process some input contacts from a sensor. The white box data make it easier to follow the various steps of the MHT algorithm. To produce this example, various "white box flags" related to the MHT algorithm would have been set to appropriate values in order to request a log of the relevant information during the operation of the MSDF Software.

•
•
•

SCAN # 2
Information read in "mht_example" measurement file:

Dimension of measurement vector = 3
Number of measurements for this scan = 3

meas #	1	2.358	2	23696.35	8.20	-0.66	0.00	0.00	0.00
meas #	2	2.381	1	24336.86	15.55	0.26	0.00	0.00	0.00
meas #	3	2.550	-1	11618.28	77.03	2.74	0.00	0.00	0.00

Function "associate_meas".

Validation matrix of each cluster:

Cluster # 1 :

Validation matrix

	measurement #		
track #	1	2	3
1	0e+00	-1	-1
	1.000	0	0
	1.000	0	0

Cluster # 2 :

Validation matrix

	measurement #		
track #	1	2	3
2	-1	0e+00	-1
	0	2.000	0
	0	1.000	0

Cluster # 3 :

Validation matrix

This new cluster does not have a validation matrix.

Function "div_cluster".

Output of the function "gen_hypotheses".

Cluster # 1 :

Number of tracks = 2
Number of hyp. = 5

Hypotheses list							
#	hypothesis tracks	M1	Ntgt	Nft	Nnt	Ndt	prob.
1	0	0	0	1	0	0	0.0000
2	T1	0	1	1	0	0	0.0000
3	T1	T1	1	0	0	1	0.9000
4	0	T4	0	0	1	0	0.0000
5	T1	T4	1	0	1	0	0.0000

Output of the function "prune".

Hypotheses list after pruning:

Cluster # 1 :

Hypotheses list							
#	hypothesis tracks	M1	Ntgt	Nft	Nnt	Ndt	prob.
1	T1	T1	1	0	0	1	1.0000

Output of the function "renumber".

Cluster # 1:

Number of tracks = 1 Number of hyp. = 1

Tracks list						
track # origin	Q	Number of Hyp.	2 last observations	Position (km)	Position (km)	Position (m)
1 (1)	7	1	[1,1]	23.4527	3.3786	-271.9566

Hypotheses list							
#	hypothesis tracks	M1	Ntgt	Nft	Nnt	Ndt	prob.
1	1	1	1	0	0	1	1.0000

Function "combine".

Output of the function "gen_hypotheses".

Cluster # 2:

Number of tracks = 2 Number of hyp. = 5

Hypotheses list							
#	hypothesis tracks	M2	Ntgt	Nft	Nnt	Ndt	prob.
1	0	0	0	1	0	0	0.0000
2	T2	0	1	1	0	0	0.0000
3	T2	T2	1	0	0	1	0.9000
4	0	T5	0	0	1	0	0.0000
5	T2	T5	1	0	1	0	0.0000

Output of the function "prune".

Hypotheses list after pruning:

Cluster # 2:

Hypotheses list							
#	hypothesis tracks	M2	Ntgt	Nft	Nnt	Ndt	prob.
1	T2	T2	1	0	0	1	1.0000

Output of the function "renumber".

Cluster # 2 :

Number of tracks = 1 Number of hyp. = 1

Tracks list						
track # origin	Q	Number of Hyp.	2 last observations	Position (km)	Position (km)	Position (m)
2 (2)	7	1	[2,2]	23.4458	6.5245	109.2711

Hypotheses list							
#	hypothesis tracks	M2	Ntgt	Nft	Nnt	Ndt	prob.
1	2	2	1	0	0	1	1.0000

Function "combine".

Output of the function "gen_hypotheses".

Cluster # 3 :

Number of tracks = 1 Number of hyp. = 2

Hypotheses list							
#	hypothesis tracks	M3	Ntgt	Nft	Nnt	Ndt	prob.
1	--	0	0	1	0	0	0.0909
2	--	T3	0	0	1	0	0.9091

CLUSTER LIST AFTER THE SCAN # 2

Cluster # 1 :

(319860)

Number of tracks = 1 Number of hyp. = 1	Prior cluster: 0 Next cluster : 320348	Tracks list: 319896 Hyp. list : 321768

Tracks list (319896)						
track # origin	Q	Number of Hyp.	2 last observations	Position (km)	Position (km)	Position (m)
1 (1)	7	1	[1,1]	23.4527	3.3786	-271.9566

Hypotheses list (321768)							
#	hypothesis tracks	M1	Ntgt	Nft	Nnt	Ndt	prob.
1	1	1	1	0	0	1	1.0000

Cluster # 2 :

(320348)

Number of tracks = 1 Number of hyp. = 1	Prior cluster: 319860 Next cluster : 321248	Tracks list: 320384 Hyp. list : 321820

Tracks list (320384)						
track # origin	Q	Number of Hyp.	2 last observations	Position (km)	Position (km)	Position (m)
2 (2)	7	1	[2,2]	23.4458	6.5245	109.2711

Hypotheses list (321820)							
#	hypothesis tracks	M2	Ntgt	Nft	Nnt	Ndt	prob.
1	2	2	1	0	0	1	1.0000

Cluster # 3 :

(321248)

Number of tracks = 1 Number of hyp. = 2	Prior cluster: 320348 Next cluster : 0	Tracks list: 321284 Hyp. list : 321892

Tracks list (321284)						
track # origin	Q	Number of Hyp.	2 last observations	Position (km)	Position (km)	Position (m)
3 (-2)	4	1	[3,0]	2.6046	11.3089	555.3979

Hypotheses list (321892)							
#	hypothesis tracks	M3	Ntgt	Nft	Nnt	Ndt	prob.
1	–	0	0	1	0	0	0.0909
2	–	T3	0	0	1	0	0.9091

SCAN # 3

•
•
•

INTERNAL DISTRIBUTION

DREV TR 1999-153

1 - Director General
1 - Deputy Director General
1 - Chief Scientist
6 - Document Library
1 - É. Bossé
1 - J.M.J. Roy (author)
1 - M. Blanchette
1 - R. Carling
1 - S. Paradis
1 - LCdr D. Morrissey
1 - R. Breton
1 - V. Pille
1 - M. Bélanger
1 - I. Abi-Zeid
1 - J. Berger
1 - A. Guitouni
1 - L. Lamontagne
1 - P. Labbé
1 - G. Vézina
1 - R. Charpentier
1 - D. Gouin
1 - G. Thibault
1 - D. Demers
1 - D. Dion
1 - L. Bissonnette
1 - L. Sévigny
1 - F. Reid
1 - Y. de Villers

INTERNAL DISTRIBUTION (cont'd)

DREV TR 1999-153

1 - J. Bédard
1 - J.-M. Garneau
1 - J.-M. Thériault
1 - J.-P. Ardouin
1 - C. Carrier
1 - J. Cruickshank
1 - J. Maheux
1 - G. Fournier

EXTERNAL DISTRIBUTION

DREV TR 1999-153

5 - DCIEM:

Attn: D. Beevis

R. Pigeau

C. McCann

S. Mcfadden

LCdr K. Kubeck

4 - DREA:

Attn: A. Ashley

B. McArthur

D. Peters

LCdr B. MacLennan

B. Chalmers

9 - DREO:

Attn: D. Liang

P. Yansouni

A. Bridgewater

C. Helleur

D. Dyck

B. Eatock

B. Ford

1 - PMO CPF

1 - PMO TRUMP

1 - PMO Aurora

1 - PMO NWS

1 - Canadian Forces Command and Staff College
Toronto
Attn: Commanding Officer

EXTERNAL DISTRIBUTION

DREV TR 1999-153

1 - DRDCIM

1 - DRDCIM (unbound copy)

1 - DRDB

1 - DMSS

4 - DMSS - 6

Attn: J. Muller

K. Chadwick

A. Beaulieu

F. Harlow

1 - DMSS - 8

1 - DMSS - 8 - 2

1 - DMSS - 8 - 8

1 - DMSS - 8 - 8 - 2

1 - DMFD

1 - DMOR

1 - DMPPD

1 - DMPPD 6

1 - DMPPD 6-2

1 - DRDCS

1 - DRDCS - 4

1 - DRDCS - 5

1 - DSAA

1 - DSAA - 3

1 - SAM

1 - DSTM

1 - DSTM - 2

EXTERNAL DISTRIBUTION

DREV TR 1999-153

1 - Canadian Forces School of Aerospace Studies
CFB Winnipeg
Westwin, Manitoba
R3J 0T0
Attn: Commanding Officer

1 - CF Maritime Warfare School
CFB Halifax
Halifax, Nova Scotia
Attn: Commanding Officer

2 - National Research Council (NRC) Canada
Institute for Information Technology
(Software Engineering Laboratory)
Ottawa, Ontario
K1A 0R6
Attn: W.M. Gentleman
C. Gauthier

1 - LCdr S.A. Midwood
CANTASS - Program Manager
DGMEPM / DMSS 7-8
ASW Sensors & Weapons
NDHQ / Louis St. Laurent Building
Hull, Quebec, K1A 0K2

1 - Maj. I.N. Glenn
UAV - Program Manager
DASPM
NDHQ / Louis St. Laurent Building
Hull, Quebec, K1A 0K2

DOCUMENT CONTROL DATA

1. ORIGINATOR (name and address) DREV 2459 Blvd Pie XI North Val-Bélair, Qc G3J 1X5	2. SECURITY CLASSIFICATION (including special warning terms if applicable) UNCLASSIFIED

3. TITLE (its classification should be indicated by the appropriate abbreviation (S, C, R or U)
Definition of a Test Bed for the Evaluation of Real-Time Multi-Source Data Fusion Systems (U)

4. AUTHORS (Last name, first name, middle initial. If military, show rank, e.g. Doe, Maj. John E.)
ROY, Jean

5. DATE OF PUBLICATION (month and year)	6a. NO. OF PAGES	6b .NO. OF REFERENCES
September 2000	90	9

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. Give the inclusive dates when a specific reporting period is covered.)
Technical Report

8. SPONSORING ACTIVITY (name and address)
N/A

9a. PROJECT OR GRANT NO. (Please specify whether project or grant)	9b. CONTRACT NO.
Project 1ba (Work Units 1ba12 and 1ba14)	N/A

10a. ORIGINATOR'S DOCUMENT NUMBER	10b. OTHER DOCUMENT NOS
DREV TR 1999-153	N/A

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

- [x] Unlimited distribution
- [ ] Contractors in approved countries (specify)
- [ ] Canadian contractors (with need-to-know)
- [ ] Government (with need-to-know)
- [ ] Defense departments

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected.)
Unlimited Announcement

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

Multi-Source Data Fusion (MSDF) is about tactical picture compilation. An MSDF system processes the data and information reported by multiple dissimilar sources in order to correctly and quickly derive the best estimates of the current and future kinematics properties for each hypothesized (or perceived) entity in the operational environment, and to develop inferences as to the identity and key attributes of these entities. From a research and development perspective, MSDF system developers need capabilities that allow them to assess if a proposed MSDF System is working properly, as expected. Tools are required to support the developer in his quantitative assessment of the performance of the MSDF algorithms and techniques. However, the MSDF systems foreseen to support command teams onboard Canadian warships must be real-time systems. The correctness of the final implementation of a real-time system is not only a function of the accuracy of the computations performed by the system, but also a function of the time at which the results of the computations are produced. This document gives a detailed description of a test bed that will help to establish the real-time requirements of MSDF systems on speed, responsiveness, predictability, timeliness, etc. At the same time, the proposed MSDF test bed will also provide a capability to evaluate various measures of performance for these MSDF Systems.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus, e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Data Fusion
Sensor Fusion
Multi-Sensor Data Fusion
Multi-Source Data Fusion
Test Bed
Performance Evaluation
Real-Time Systems

#514219